

SCHOOL OF COMPUTING & INFORMATION SCIENCES

**Annual Assessment Summary 2017-2019
for the
Bachelor of Science in Computer Science**

Prepared by
Jainendra Navlakha – Assessments Coordinator

November 27, 2019

Table of Contents

- I. INTRODUCTION
- II. OVERVIEW
 - A. **Terminology**
 - B. **Assessment Mechanisms & Procedures**
 - C. **Process**
- III. DATA
 - A. **Course Outcomes Survey by Students**
 - B. **Course Outcomes Survey by Instructors**
 - C. **Graduating Student (Exit) Survey of Student Outcomes**
 - D. **Alumni Survey of Program Educational Objectives**
 - E. **Employer Survey of Program Educational Objectives**
 - F. **Course Embedded Assessment**
 - G. **Capstone Project Assessment**
- IV. EVALUATION
 - A. **Course Outcomes**
 - B. **Student Outcomes**
 - C. **Program Educational Objectives**
- V. RECOMMENDATIONS
 - A. **Recommendations of the Subject Area Coordinators**
 - B. **Recommendations of the Assessments Coordinator**
- VI. CONCLUSION
- VII. APPENDICES

I. INTRODUCTION

This report is prepared in accordance with the intent of the Assessment Plan originally adopted by the School of Computing & Information Sciences (then the School of Computer Science) in spring 2003, and last version approved in spring 2015. Its purpose is to summarize the results of the various assessment mechanisms utilized by SCIS in support of the BS in Computer Science program, and to present the resulting findings and recommendations to the Undergraduate Committee, the Undergraduate Program Director, the Faculty of the School, and the Director.

Since the last Assessment cycle (Summer 2015 to Spring 2017), no modifications are made to the Assessment Mechanisms and Procedures.

The goals of the assessment process are to assess the extent to which the Student Outcomes and Program Educational Objectives of the BS in Computer Science program have been attained in the period under review, to identify specific areas of the program where a need for improvement is indicated, and to present a set of recommendations for achieving those improvements.

This review is conducted for the period from Summer 2017 to Spring 2019.

II. OVERVIEW

A. Terminology

The BS in Computer Science *Program Educational Objectives* (Appendix A-1) document describes the overriding goals of the program relating to the cumulative persistent effects of the students' educational experiences. The objectives are broad in nature and define expected general characteristics of the program's graduates within some years after graduation.

The BS in Computer Science *Student Outcomes* (Appendix A-2) are more specific in nature. These describe characteristics of students at the time of graduation, and define the specific knowledge, skills, and behaviors that they are expected to acquire as they complete the requirements of the program. Attainment of each Student Outcome enables the attainment of one or more of the Program Educational Objectives.

Do note that the Program Educational Objectives were reorganized in the last assessment cycle, and the new set became effective in Fall 2015. The Student Outcomes were rewritten (mostly, reorganized) to match those prescribed by ABET, and the new ones also became effective in Fall 2015. No modifications are made to those POs and SOs in this assessment cycle.

The syllabus of each required and elective course of the BS in Computer Science program presents a set of *Course Outcomes*. The Course Outcomes identify specific knowledge units and levels of attainment (mastery, familiarity, awareness) expected of a student completing the course. Attainment by students of Course Outcome enables attainment of one or more of the Student Outcomes.

B. Assessment Mechanisms & Procedures

Consistent with current educational practice, SCIS follows a systematic process of collecting and utilizing data on the degree of attainment of the Student Outcomes and Program Educational Objectives. The *SCIS Assessment Plan* (Appendix B-1) specifies the participants and schedule for this process, and the means of evaluating the data and enacting program changes indicated by the evaluation. The *SCIS Assessment Mechanisms & Procedures* document (Appendix B-2) specifies the implementation of the Assessment Plan. The SCIS Assessment Plan and Assessment Procedures and Mechanisms were adopted in 2003, amended in 2010 to incorporate additional direct assessment measures, and last amended in spring 2015 to align better with our changed operations.

The following indirect assessment mechanisms have been employed in this assessment cycle:

<u>Mechanism</u>	<u>Target</u>	<u>Frequency</u>
Course Outcomes Survey by Students	Course Outcomes	Semester
Course Outcomes Survey by Instructors	Course Outcomes	Semester
Graduating Student (Exit) Survey	Student Outcomes	Semester
Alumni Survey	Program Educational Objectives	Continual
IAB Members and Employers Survey	Program Educational Objectives	Continual

The following direct assessment mechanisms have been employed since spring 2010:

<u>Mechanism</u>	<u>Target</u>	<u>Frequency</u>
Course Embedded Assessment	Course Outcomes and Student Outcomes	At least once in the Assessment Period
Capstone Project Assessment	Student Outcomes	Semester

Additional input is solicited and may be received from other program constituents including:

- ACM Student Chapter,
- Upsilon Pi Epsilon Honor Society Chapter,
- SCIS Women In Computer Science group,
- STARS Student Chapter,
- Programming Team, and
- SCIS Industry Advisory Board.

C. Process

The required and elective courses of the BS in Computer Science are each assigned, based on subject area, to one of seven groups: Applications (new group created in the current cycle), Computer Organization, Computer Systems, Foundations, Professional Development, Programming, and Software Engineering.

Each subject area group is managed by a faculty Subject Area Coordinator (SAC). Periodically, the assessment data and comments from Student and Instructor Course Outcome Surveys are considered by the Subject Area Coordinators. These provide the information for the Subject Area Coordinators' reports.

The SAC reports and assessment data from all other sources are evaluated by the SCIS Assessments Coordinator whose evaluations and recommendations are presented in an assessment report.

The assessment report is considered by the SCIS Undergraduate Committee, and by the SCIS Undergraduate Program Director. The Undergraduate Committee's curricular recommendations are presented to the SCIS faculty for approval. Responsibility for enactment of approved recommendations rests with the SCIS Undergraduate Program Director.

III. DATA

A. Course Outcomes Survey by Students

This survey is completed by students in each section of a required or elective CS class. For each course outcome, the student states the extent to which (s)he agrees with the following two assertions:

- 1: *I believe that this is a valuable outcome for this course, and*
 2: *The subject matter of this outcome was covered adequately in class*

To each assertion, the student responds on a 5-point scale as follows:

- 5: *I agree strongly, 4: I agree moderately, 3: I am not sure, 2: I disagree moderately, 1: I disagree strongly*

For each outcome, a weighted mean of the responses to each question is calculated. The means are provided for each course, cumulatively over all semesters of the period under review.

	<u>BS in Computer Science Required or Elective Course</u>	<u># Responding</u>	<u>Value of Outcome</u>	<u>Coverage Adequacy</u>
CAP 4104	Human-Computer Interaction	58	4.90	4.80
CAP 4630	Artificial Intelligence	24	3.79	3.38
CAL 4641	Natural Language processing	36	4.92	4.78
CAP 4710	Principles of Computer Graphics	1	4.88	4.12
CAP 4770	Introduction to Data Mining	36	4.9	4.59
CDA 3103	Fundamentals of Computer Systems	82	4.24	4.00
CDA 4101	Structured Computer Organization	64	4.40	4.18
CDA 4625	Introduction to Mobile Robotics	24	4.79	3.92
CEN 4010	Software Engineering I	58	4.78	4.46
CEN 4021	Software Engineering II	17	4.87	4.90
CEN 4072	Software Testing	42	4.58	4.21
CEN 4083	Cloud Computing	2	3.88	3.25
CGS 1920	Introduction to Computing	39	4.44	4.48
CGS 3095	Technology in the Global Arena	119	4.58	4.51
CIS 4911	Senior Project	90	4.69	4.26
CNT 4713	Net-Centric Computing	102	4.73	4.40
COP 2210	Computer Programming I	173	4.61	4.39
COP 3337	Computer Programming II	123	4.42	4.08
COP 3530	Data Structures	97	4.55	4.09
COP 4226	Advanced Windows Programming	17	4.49	4.52
COP 4338	Computer Programming III	96	4.58	4.09
COP 4520	Introduction to Parallel Computing	7	4.83	4.62
COP 4534	Algorithm Techniques	20	4.64	4.20
COP 4555	Principles Programming Languages	45	4.44	4.39
COP 4604	Advanced Unix Programming	NO	DATA	AVAILABLE
COP 4610	Operating Systems Principles	90	4.69	4.44
COP 4710	Database Management	75	4.76	4.45
COP 4722	Survey of Database Systems	42	4.45	3.95
COT 3100	Discrete Structures	78	4.32	4.34
COT 3541	Logic for Computer Science	42	4.75	4.80

COT 4521	Introduction to Comp. Geometry	NO	DATA	AVAILABLE
MAD 2104	Discrete Mathematics			
MAD 3512	Theory of Algorithms			
		=====	=====	=====
		1701	4.58	4.34

Table 1: Value & Adequacy of Coverage of Course Outcomes 05/17 – 04/19

Notes: (1) In this assessment cycle, MAD 2104 is replaced by COT 3100. MAD 2104 was taught by the Department of Mathematics; COT 3100 is taught by SCIS, and hence, for the first time, we have Student Course Outcomes available for this required course. (2) *The overall scores for Value of Outcomes (4.58) and Coverage Adequacy (4.34) are essentially the same as found in the last Assessment Report (4.63 and 4.52 respectively).* (3) COT 4521 and COP 4604 were taught only once each during the period of this assessment, but no data is available.

The semester data for each course are presented here grouped under the seven subject areas. The Subject Area Coordinator (SAC) reports are included in Appendix C.

Subject Area: Applications – [NEW] -- (SAC: Mark Finlayson)

CAP 4104 Human-Computer Interaction

CAP 4453 Introduction to Robot Vision.

This course was not offered during the evaluation period.

CAP 4630 Artificial Intelligence

CAP 4641 Natural Language Processing

CDA 4625 Introduction to Mobile Robotics

CAP 4104 – Human-Computer Interaction

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SPR 2018	37	4.84	4.68
SPR 2019	21	5.00	5.00
	=====	=====	=====
	58	4.90	4.80

Table 2-CAP 4104: Student Rating of Course Outcomes

CAP 4630 – Artificial Intelligence

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
FALL 2017	24	3.79	3.38
	=====	=====	=====

24 3.79 3.38

Table 2-CAP 4630: Student Rating of Course Outcomes

CAP 4641 – Natural Language Processing

	<u># Responding</u>	<u>Value of Outcome</u>	<u>Coverage Adequacy</u>
SPR 2018	32	4.91	4.75
SPR 2019	4	5.00	5.00
	=====	=====	=====
	36	4.92	4.78

Table 2-CAP 4641: Student Rating of Course Outcomes

CDA 4625 – Introduction to Mobile Robotics

	<u># Responding</u>	<u>Value of Outcome</u>	<u>Coverage Adequacy</u>
SPR 2018	24	4.79	3.92
	=====	=====	=====
	24	4.79	3.92

Table 2-CDA 4625: Student Rating of Course Outcomes

Subject Area: Professional Development (SAC: Richard Whitaker)

CGS 1920 Introduction to Computing

CGS 3095 Technology in the Global Arena

ENC 3249 Professional and Technical Writing for CS (Taught by English Department)

CGS 1920 -- Introduction to Computing

	<u># Responding</u>	<u>Value of Outcome</u>	<u>Coverage Adequacy</u>
SUM 2017	8	4.77	4.80
FALL 2017	6	4.36	4.38
SPR 2018	16	4.15	4.20
SUM 2018	DATA NOT AVAILABLE		
Fall 2018	2	4.29	4.50
SPR 2019	7	4.84	4.83
	=====	=====	=====
	39	4.44	4.48

Table 2-CGS 1920: Student Rating of Course Outcomes

CGS 3095 -- Prof. Ethics & Social Issues in Computing

	<u># Responding</u>	<u>Value of Outcome</u>	<u>Coverage Adequacy</u>
SUM 2017	39	4.38	4.21
FALL 2017	22	4.77	4.61
SPR 2018	38	4.73	4.77
SUM 2018	NA		
FALL 2018	13	4.53	4.54
SPR 2019	7	4.30	4.44
	=====	=====	=====
	119	4.58	4.51

Table 2-CGS 3095: Student Rating of Course Outcomes

Subject Area: Computer Organization (SAC: Nagarajan Prabakar)

CDA 3102 Computer Architecture

New course to replace CDA 3103 and CDA 4101 starting in Spring 2020

CDA 3103 Fundamentals of Computer Systems

CDA 4101 Structured Computer Organization

CNT 4713 Net-centric Computing

COP 4610 Operating Systems Principles

CDA 3103 -- Fundamentals of Computer Systems

	<u># Responding</u>	<u>Value of Outcome</u>	<u>Coverage Adequacy</u>
SUM 2017	13	4.65	4.62
FALL 2017	27	4.00	3.80
SPR 2018	19	4.44	3.86
SUM 2018	6	4.52	4.63
FALL 2018	11	4.17	3.91
SPR 2019	6	3.72	3.61
	=====	=====	=====
	82	4.24	4.00

Table 2-CDA 3103: Student Rating of Course Outcomes

CDA 4101 -- Structured Computer Organization

	<u># Responding</u>	<u>Value of Outcome</u>	<u>Coverage Adequacy</u>
SUM 2017	NA		
FALL 2017	23	4.38	4.04
SPR 2018	26	4.57	4.51
SUM 2018	3	4.27	4.53

FALL 2018	8	4.30	3.60
SPR 2019	4	3.65	3.65
	=====	=====	=====
	64	4.40	4.18

Table 2-CDA 4101: Student Rating of Course Outcomes

CNT 4713 – Net-Centric Computing

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SUM 2017	14	4.90	4.87
FALL 2017	37	4.83	4.72
SPR 2018	36	4.49	3.75
SUM 2018	3	5.00	4.85
FALL 2018	5	4.89	4.66
SPR 2019	7	4.96	4.75
	=====	=====	=====
	102	4.73	4.40

Table 2-CNT 4713: Student Rating of Course Outcomes

COP 4610 -- Operating Systems Principle

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SUM 2017	14	4.46	4.04
FALL 2017	28	4.78	4.61
SPR 2018	23	4.79	4.56
SUM 2018	8	4.50	4.15
FALL 2018	10	4.80	4.48
SPR 2019	7	4.54	4.37
	=====	=====	=====
	90	4.69	4.44

Table 2-COP 4610: Student Rating of Course Outcomes

Subject Area: Computer Systems (SAC: Jason Liu)

CAP 4612 Introduction t Machine Learning

[The course was not offered during the evaluation period]

CAP 4710 Principles of Computer Graphics

CAP 4770 Introduction to Data Mining

CEN 4083 Cloud Computing

COP 4604 Advanced UNIX Programming

COP 4710 Database Management systems

COP 4722 Survey of Database Systems

CAP 4710 – Principles of Computer Graphics

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SPR 2019	1	4.88	4.12
	=====	=====	=====
	1	4.88	4.12

Table 2-CAP 4710: Student Rating of Course Outcomes

CAP 4770 -- Principles of Data Mining

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
FALL 2017 – RVC*	10	4.93	4.75
SPR 2018 – RVC	12	4.90	4.35
SUM 2018 – RVAA#	6	5.00	4.94
FALL 2018 - RVC	3	4.44	4.00
SPR 2019 - RVC	5	5.00	4.80
	=====	=====	=====
	36	4.90	4.59

*RVC – Online Course

#RVAA - Special Online

Table 2-CAP 4770: Student Rating of Course Outcomes

CEN 4083 – Cloud Computing

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
FALL 2018	2	3.88	3.25
	=====	=====	=====
	2	3.88	3.25

Table 2-CEN 4083: Student Rating of Course Outcomes

COP 4604 -- Advanced UNIX Programming

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SUM 2017	NA		
	=====	=====	=====

Table 2-COP 4604: Student Rating of Course Outcomes

COP 4710 -- Database Management Systems

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SUM 2017	1	4.71	4.86
FALL 2017	9	4.86	4.84
FALL 2017 – RVC	9	4.67	3.60
FALL 2017	6	4.62	4.24
SPR 2018	15	4.99	4.90
SPR 2018	3	4.71	4.24
SPR 2018	2	4.43	4.43
SPR 2018 - RVC	10	4.89	4.59
SUM 2018	4	4.68	4.79
FALL 2018	1	4.86	4.86
FALL 2018	3	4.90	4.95
FALL 2018	1	3.86	3.86
FALL 2018 – RVC	4	4.82	4.79
SPR 2019	1	5.00	5.00
SPR 2019	3	4.95	4.95
SPR 2019 – RVC	3	3.57	1.67
	=====	=====	=====
	75	4.76	4.45

Table 2-COP 4710: Student Rating of Course Outcomes

COP 4722 -- Survey of Database Systems

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SUM 2017 – RVAA	NA		
FALL 2017	5	4.88	3.84
FALL 2017 – RVC	8	4.22	4.18
SPR 2018	5	3.84	3.32
SPR 2018	2	5.00	5.00
SPR 2018 – RVC	10	4.22	3.66
SUM 2018 – RVAA	3	5.00	4.27
FALL 2018	5	4.44	3.52
FALL 20168 – RVC	1	5.00	5.00
SPR 2019	1	5.00	4.80
SPR 2019 – RVC	2	5.00	5.00
	=====	=====	=====
	42	4.45	3.95

Table 2-COP 4722: Student Rating of Course Outcomes

Subject Area: Foundations (SAC: Xudong He)

CAP 4506 Introduction to Game Theory [NEW]

[This course was offered in Spring 2019 – No course evaluations were submitted]

COP 4534 Algorithm Techniques

COP 4555 Principles of Programming Languages

COT 3100 Discrete Structures [NEW]

COT 3541 Logic for Computer Science

COT 4521 Introduction to Computational Geometry

[One section was offered in Fall 2018 – No student evaluations were submitted.]

MAD 3512 Introduction to Theory of Algorithms (No data is available)

Set 2 (Math) Electives (**MAD 3305, MAD 3402, MAD 4203, MHF 4302**)

COP 4534 – Algorithm Techniques

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
FALL 2017	9	4.56	4.15
SPR 2018	7	4.62	4.24
FALL 2018	2	5.00	5.00
SPR 2019	2	4.67	3.50
	=====	=====	=====
	20	4.64	4.20

Table 2-COP 4534: Student Rating of Course Outcomes

COP 4555 -- Principles of Programming Languages

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
FALL 2017	9	4.37	4.35
FALL 2017	12	4.32	4.10
SPR 2018	6	4.28	4.25
SPR 2018	5	4.70	4.53
SPR 2018	2	4.58	4.58
FALL 2018	1	5.00	5.00
FALL 2018	1	4.17	4.50
SPR 2019	5	4.73	4.86
SPR 2019	2	4.92	5.00
SPR 2019	1	5.00	5.00
SPR 2019	1	2.83	3.00
	=====	=====	=====
	45	4.44	4.39

Table 2-COP 4555: Student Rating of Course Outcomes

COT 3100 – Discrete Structures

<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
-------------------------------	-----------------------------------	------------------------------------

SUM 2017	2	4.00	4.00
SUM 2017	4	4.33	4.25
FALL 2017	7	4.35	4.31
FALL 2017	1	1.83	1.83
FALL 2017	3	4.47	4.53
FALL 2017	2	2.71	3.00
FALL 2017	5	3.45	3.50
SPR 2018	2	5.00	4.93
SPR 2018	2	4.71	4.93
SPR 2018	12	4.40	4.56
SPR 2018	8	4.51	4.23
SPR 2018	6	4.55	4.57
SPR 2018	3	4.05	3.52
FALL 2018	3	5.00	5.00
FALL 2018	1	4.14	4.43
FALL 2018	1	4.14	5.00
FALL 2018	1	5.00	4.86
FALL 2018	2	4.29	4.43
FALL 2018	1	4.75	5.00
SPR 2019	3	4.52	4.52
SPR 2019	2	4.21	4.21
SPR 2019	1	5.00	5.00
SPR 2019	3	4.10	4.71
SPR 2019	3	5.00	4.71
	=====	=====	=====
	78	4.32	4.34

Table 2-COT 3100: Student Rating of Course Outcomes

COT 3541 -- Logic for Computer Science

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SUM 2017	2	5.00	4.62
FALL 2017	8	4.75	4.84
FALL 2017	9	4.66	4.74
FALL 2017	2	4.50	4.50
SPR 2018	5	4.60	4.90
SPR 2018	2	4.61	4.79
SPR 2018	4	5.00	4.62
FALL 2018	3	4.58	4.75
FALL 2018	4	5.00	5.00
FALL 2018	1	4.75	5.00
SPR 2019	2	5.00	5.00
	=====	=====	=====
	42	4.75	4.80

Table 2-COT 3541: Student Rating of Course Outcomes

Set 2 (Math) Electives

MAD 3305 Graph Theory

MAD 3402 Numerical analysis

MAD 4203 Introduction to Combinatorics

MHF 4302 Mathematical Logic

The Set 2 Elective courses are taught by faculty of the Mathematics Department. There are no assessment data available for these courses.

Subject Area: Programming (SAC: Tim Downey)

COP 2210 Computer Programming I

COP 3337 Computer Programming II

COP 3530 Data Structures

COP 4226 Advanced Windows Programming

COP 4338 Computer Programming III

COP 4520 Introduction to Parallel Computing

COP 2210 – Computer programming I

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SUM 2017	57	4.73	4.64
FALL 2017	4.	4.46	4.07
SPR 2018	21	4.46	4.21
SUM 2018	26	4.85	4.75
FALL 2018	19	4.34	4.12
SPR 2019	10	4.74	4.26
	=====	=====	=====
	173	4.61	4.39

Table 2-COP 2210: Student Rating of Course Outcomes

COP 3337 -- Computer Programming II

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SUM 2017	40	4.64	4.16
FALL 2017	23	4.64	4.42
SPR 2018	27	4.35	4.08
SUM 2018	10	4.40	4.11
FALL 2018	8	4.24	3.66
SPR 2019	15	3.71	3.54
	=====	=====	=====

123

4.42

4.08

Table 2-COP 3337: Student Rating of Course Outcomes

COP 3530 -- Data Structures			
	<u>#</u>	<u>Value of</u>	<u>Coverage</u>
	<u>Responding</u>	<u>Outcome</u>	<u>Adequacy</u>
SUM 2017	6	4.48	4.43
FALL 2017	32	4.63	4.49
SPR 2018	27	4.40	4.23
SUM 2018	11	4.28	4.02
FALL 2018	13	4.73	4.76
SPR 2019	8	4.88	4.73
	=====	=====	=====
	97	4.55	4.42

Table 2-COP 3530: Student Rating of Course Outcomes

COP 4226 -- Advanced Windows Programming			
	<u>#</u>	<u>Value of</u>	<u>Coverage</u>
	<u>Responding</u>	<u>Outcome</u>	<u>Adequacy</u>
FALL 2017	13	4.47	4.53
FALL 2018	4	4.56	4.50
	=====	=====	=====
	17	4.49	4.52

Table 2-COP 4226: Student Rating of Course Outcomes

COP 4338 -- Computer Programming III			
	<u>#</u>	<u>Value of</u>	<u>Coverage</u>
	<u>Responding</u>	<u>Outcome</u>	<u>Adequacy</u>
SUM 2017	14	4.55	4.64
FALL 2017	33	4.58	4.18
SPR 2018	22	4.68	3.84
SUM 2018	9	4.43	3.62
FALL 2018	14	4.53	4.02
SPR 2019	4	4.59	4.16
	=====	=====	=====
	96	4.58	4.09

Table 2-COP 4338: Student Rating of Course Outcomes

COP 4520 -- Introduction to Parallel Computing			
	<u>#</u>	<u>Value of</u>	<u>Coverage</u>

	<u>Responding</u>	<u>Outcome</u>	<u>Adequacy</u>
SPR 2018	5	4.80	4.63
SPR 2019	2	4.92	4.58
	=====	=====	=====
	7	4.83	4.62

Table 2-COP 4520: Student Rating of Course Outcomes

Subject Area: Software Engineering (SAC: Masoud Sadjadi)

CEN 4010 Software Engineering I

CEN 4021 Software Engineering II

CEN 4072 Software Testing

CIS 4911 Senior Project

IDS 4918 VIP Program – [Essentially CIS 4911 for non-majors – Data collected with CIS 4911]

CEN 4010 -- Software Engineering I

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SUM 2017	6	4.62	4.06
FALL 2017	25	4.88	4.58
SPR 2018	18	4.81	4.66
FALL 2018	5	4.74	4.28
SPR 2019	4	4.25	3.59
	=====	=====	=====
	58	4.78	4.46

Table 2-CEN 4010: Student Rating of Course Outcomes

CEN 4021 -- Software Engineering II

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
FALL 2017	8	4.75	4.81
SPR 2018	3	5.00	5.00
FALL 2018	4	4.94	4.94
SPR 2019	2	5.00	5.00
	=====	=====	=====
	17	4.87	4.90

Table 2-CEN 4021: Student Rating of Course Outcomes

CEN 4072 -- Software Testing

	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SUM 2017	4	5.00	4.96
FALL 2017	8	4.55	4.25
SPR 2018	16	4.31	3.68
SUM 2018	7	4.98	4.93
FALL 2018	7	4.57	4.20
	=====	=====	=====
	42	4.58	4.21

Table 2-CEN 4072: Student Rating of Course Outcomes

CIS 4911 -- Senior Project			
	<u>#</u> <u>Responding</u>	<u>Value of</u> <u>Outcome</u>	<u>Coverage</u> <u>Adequacy</u>
SUM 2017	31	4.67	4.50
FALL 2017	21	4.78	4.33
SPR 2018	11	4.73	4.19
SUM 2018	2	4.73	4.09
FALL 2018	16	4.56	3.67
SPR 2019	9	4.72	4.44
	=====	=====	=====
	90	4.69	4.26

Table 2-CIS 4911: Student Rating of Course Outcomes

B. Course Outcomes Survey by Instructors

This survey, called the Instructor Course Appraisal (ICA), is completed by each instructor of a required or elective CS course section.

- The Instructor separately rates the individual course outcomes in respect of two criteria
 Appropriateness: *Essential* *Very Appropriate* *Appropriate* *Inappropriate*
 Coverage: *Extensive* *Adequate* *Not Enough* *Not At All*
- The Instructor separately rates the course prerequisites in respect of two criteria
 Relevance: *Irrelevant* *Incidental* *Useful* *Highly Useful*
 Student Mastery: *Good* *Adequate* *Deficient* *Non-existent*
- The Instructor rates the students' overall preparation for taking the course
 Student Preparation: *Good* *Adequate* *Deficient* *Non-existent*
- In addition, the Instructor may append general comments and suggestions specific to each course prerequisite or outcome.

These responses, comments and suggestions from the ICAs, together with the data from the Student Course Outcomes surveys (see **Table 1**) and student comments, form the basis of the Subject Area Coordinators' reports. The summaries included in this section are mostly based on these SAC reports, with occasional augmentation directly from the ICAs. As noted in the preceding section, the complete SAC reports from which these observations are taken are included in Appendix C.

Note: The data here are qualitative; no numeric scores are assigned to responses.

Subject Area: Applications (SAC: Mark Finlayson)

CAP 4104 Human-Computer Interaction

- Students agreed with the overall Valuation of Outcomes as well as their Coverage strongly (4.90 and 4.80 respectively).
- Some students mentioned that this class should be mandatory to take with Software Engineering I. While the suggestion to make this class required is an interesting one, this goes against the recent changes in the SCIS curriculum to remove requirements so as to give students more flexibility.
- Instructor evaluations included:
 - ❖ Student preparation was good
 - ❖ All objectives were essential, except for one that was appropriate

CAP 4453 Introduction to Robot Vision

- This course was not offered during the evaluation period. Therefore, no data is available to make recommendations for modification of the course.

CAP 4630 Artificial Intelligence

- In the one section that was offered during the evaluation period, students agreed with the overall Valuation of Outcomes moderately (3.79) and with their Coverage, in an average way (3.38).
- The SAC report states, "Student reactions to this class were negative, with at least five students (20% of the class) strongly disagreeing of the importance of the overall value of the outcomes and the overall adequacy of coverage. Student negative reactions seemed to have a lot to do with poor teaching by the instructor (for example, reading long text-heavy slide decks), as well as the amount of homework that was assigned and the length of the exams, with several students suggesting the amounts were either completely or very unreasonable. At least three free text comments suggested that the course covered too much material, and I agree after my own review of the course syllabus. In particular, the course includes a large unit on game theory and multi-agent systems, which strikes me as inappropriate for an introductory undergraduate class on AI."
- Instructor evaluations included:
 - ❖ Student preparation was adequate
 - ❖ All objectives were essential

CAP 4641 Natural Language Processing

- Students agreed with the overall Valuation of Outcomes as well as their Coverage strongly (4.92 and 4.77 respectively).
- Instructor evaluations included:
 - ❖ Student preparation was adequate
 - ❖ All objectives were essential

CDA 4625 Introduction to Mobile Robotics

- In the one section that was offered during the evaluation period, students agreed with the overall Valuation of Outcomes strongly (4.79) and with their Coverage, moderately (3.92).
- These specific student comments included:
 - ❖ More hands on with robots. Most was conceptual but there was really a lack of application. There should be a regular lab working on robots (especially towards the final robot project).
 - ❖ Have more YouTube videos and pictures showing examples of the material.

Subject Area: Professional Development (SAC: Richard Whitaker)

CGS1920 Introduction to Computing

- Students agreed with the overall Valuation of Outcomes as well as their Coverage either strongly or moderately (4.44 and 4.48 respectively).
- The faculty that have taught this course have discussed changing the title of this course to “Introduction to the Field of Computing”. In the past, it has been brought up to change the title to "Seminar in Computing" to clarify that it is not a programming course. Currently, the faculty believes that “Introduction to the Field of Computing” would be a better choice.

CGS 3095 Technology in the Global Arena

- Students agreed with the overall Valuation of Outcomes as well as their Coverage either strongly or moderately (4.58 and 4.51 respectively).
- The majority of students found the course material beneficial and adequate for understanding key computing related issues.
- Some students requested that the course material should focus more on the impacts of social media and destructiveness of tech startups.
- In addition, a few students commented that the textbook was not helpful for the course.

ENC 3249 Professional and Technical Writing

- There was no CES Assessment data for this course – the course is taught by the English department.
- Using the CGS 3095 course which has writing assignments as a proxy, students’ writing skills were found to range from deficient to adequate.

Subject Area: Computer Organization (SAC: Nagarajan Prabakar)

CDA 3102 Computer Architecture

- CDA-3102 is a new course to replace CDA-3103 and CDA-4101. Since CDA-3102 will be offered only from Spring 2020, there is no evaluation for this course.

CDA 3103 Fundamentals of Computer Systems

- For all five outcomes of the course, most of the students (more than 80%) agree either strongly or moderately.
- There is no significant concern expressed in the Students Suggestions section.

CDA 4101 Structured Computer Organization

- For all five outcomes of the course, most of the students (more than 80%) agree either strongly or moderately.
- Students expressed a big learning curve in writing Verilog code for designs.
- Students expressed concern about sharing the work in group projects and the credit for each team member.
- There is no significant concern expressed by the students or faculty.

CNT 4713 Net-centric Computing

- For all seven outcomes of the course, most of the students (more than 80%) agree either strongly or moderately.
- There is no significant concern expressed by the students or faculty.

COP 4610 Operating Systems Principles

- For all five outcomes of the course, most of the students (more than 80%) agree either strongly or moderately.
- There is no significant concern expressed by the students or faculty.

Subject Area: Computer Systems (SAC: Jason Liu)

CAP 4612 Introduction to Machine Learning

- This course was not offered during the evaluation period.

CAP 4710 Principles of Computer Graphics

- In the one section that was offered during the evaluation period, students agreed with the overall Valuation of Outcomes strongly (4.88) and with their Coverage, moderately (4.18).
- The instructor did not submit the course appraisal for the session.

CAP 4770 Principles of Data Mining

- This course has six outcomes, all of which have been indicated by the instructors as either essential or appropriate.
- Students agreed with the overall Valuation of Outcomes strongly (4.90) and with their Coverage, strongly/moderately (4.59).

CEN 4083 Cloud Computing

- In the one section that was offered during the evaluation period, students agreed with the overall Valuation of Outcomes moderately (3.88) and with their Coverage, in an average way (3.25).
- This course has four outcomes, all of which have been indicated by the instructor as essential.

COP 4604 Advanced UNIX Programming

- This course was taught only once during this evaluation period.
- The instructor didn't submit the course appraisal for this session.
- The student evaluation for this session (only one evaluation received) is available in the system, but it did not include the evaluation of Course Outcomes.
- This course has six outcomes, all of which have been indicated by the instructor as essential or appropriate.

COP 4710 Database Management systems

- Students agreed with the overall Valuation of Outcomes strongly (3.79) and with their Coverage, moderately (4.45).
- Summary of Assessment: This course has seven outcomes, all of which have been indicated by the instructors as either essential or appropriate.

COP 4722 Survey of Database Systems

- Students agreed with the overall Valuation of Outcomes as well as their Coverage moderately (4.45 and 3.95 respectively).
- This course has five outcomes. One instructor indicated that all the outcomes are essential, very appropriate, or appropriate. However, another instructor consistently indicated that the objective "Object-Oriented Database" and "Spatial and Multimedia Databases" as inappropriate.

Subject Area: Foundations (SAC Xudong He)

CAP 4506 Introduction to Game Theory

- This course was only offered once in the evaluation period.
- Three students submitted course evaluations, but did not provide any answers on Course Outcomes.
- The only student suggestion was "would like to have more homework and projects to cement the concepts".
- The instructor did not provide any comments or suggestions in course appraisal.

COP 4534 Algorithm Techniques

- Students agreed with the overall Valuation of Outcomes strongly (4.64) and with their Coverage, moderately (4.20).
- Most students' comments were on homework assignments. Some student felt the homework assignments were very rewarding and challenging. Several students felt more homework assignment were needed in Fall 2017; however one student comment in 2019

suggested to reduce the number of homework by one. The instructor adjusted the number of assignments during the past two years.

- A few student comments in Fall 2017 were about more discussion and review for exams and making lectures more organized and engaged.
- The instructor's comment on student preparation went from deficient in Fall 2017 to adequate afterwards, and suggested that students should have some basic knowledge of combinatorics, statistics, and probability before taking this course.

COP 4555 Principles of Programming Languages

- Students agreed with the overall Valuation of Outcomes as well as their Coverage strongly/moderately (4.44 and 4.39 respectively). The overall student responses were low (many classes had only 1 or 2).
- Student comments included to have more variation of practice exercises, to have a textbook, to have more quizzes, to have extra points for students willing to put in extra effort, to provide solutions for homework assignments.
- Instructors' evaluations indicate that the students' preparation for this course ranges from adequate to good.
- Only a few instructor appraisal comments stated that students need better mathematics preparation to understand the essential concepts of functions, sets, and relations; better rigorous thinking and logical reasoning capabilities; and that the course be taught in a laboratory to practice programming in F#.

COT 3100 Discrete Structures

- Students agreed with the overall Valuation of Outcomes as well as their Coverage moderately (4.32 and 4.34 respectively). The overall student responses were low (many classes had only 1 to 3).
- Student comments included to have more homework assignments and in class practice, to provide some tutoring, and to use a better textbook in some section.
- Overall the students felt this was a challenging course.
- Students' preparation for this course ranges from non-existent, deficient, adequate to good.
- Only a few instructor appraisal comments included
 - ❖ students must develop stronger work ethics prior to enrolling in this course
 - ❖ the number of the objectives is too high
 - ❖ compress outcomes related to programming into a single outcome and make it be "familiarity" rather than implementation
 - ❖ students have a very low level of math and logical reasoning, and therefore it is very difficult for them to formalize problems and proofs
 - ❖ there is no time to properly cover some of the objectives related to program implementation.

COT 3541 Logic for Computer Science

- Students agreed with the overall Valuation of Outcomes as well as their Coverage strongly (4.75 and 4.80 respectively). Overall student responses were low, in single digits for all eleven sections taught during this period.
- Student comments include:
 - ❖ to connect logic to real world applications

- ❖ to have homework graded or provide answers
- ❖ to have more consistency among the professors teaching the same course
- ❖ to have quick email response to student questions
- ❖ to have videos for explaining course materials
- ❖ to have more time on Prolog, to have a better textbook
- ❖ to have more examples.
- ❖ One comment of the online offering was to change discussion posts to classwork.
- Students' preparation for this course was adequate.
- Only a few instructor appraisal comments stated that this course has effectively challenged students to think and logic provides the unifying foundation for computer science. One suggestion was to explicitly cover propositional logic to help students have a consistent and systematic knowledge of various concepts in logic.

COT 4521 Introduction to Computational Geometry

- For the one section taught during the evaluation period, no student evaluations are available.
- The instructor commented that the overall student preparation for this course was good, but additional prerequisites such as linear algebra and programming could be helpful, and using more demos could also help student understanding.

MAD 2104 Discrete Mathematics

- Essentially, substituted by COT 3100 in this assessment cycle.

MAD 3512 Theory of Algorithms

- Taught by Mathematics department. Neither the instructor appraisals nor student evaluations are available.

Set 2 (Math) Electives (MAD 3305, MAD 3402, MAD 4203, MHF 4302)

- *(Assessments Coordinator :) These courses are taught by the Math department faculty and consequently are not subject to the School's assessment mechanisms.*

Subject Area: Programming (SAC: Tim Downey)

COP 2210 Computer Programming I

- Students agreed with the overall Valuation of Outcomes as well as their Coverage strongly/moderately (4.61 and 4.39 respectively).
- From instructor course appraisals, students seem to be deficient in mathematical preparation for the course. Some instructors want a math prerequisite, others want a programming prerequisite. Since the time of these comments, a prerequisite of pre-calculus has been added to the course.
- Other comments are varied: enforce objects first; do not cover arrays, only cover array list; limit enrollment to CS majors, create a problem-solving prerequisite; require a lab or loaner laptops that can be kept throughout the semester.

COP 3337 Computer Programming II

- Students agreed with the overall Valuation of Outcomes as well as their Coverage moderately (4.42 and 4.08 respectively).
- From instructor course appraisals, students seem to be deficient in several of the prerequisite outcomes: methods and parameters; selection and iteration; String, ArrayList and Wrappers. One instructor noted a deficiency in all the prerequisite outcomes. An online instructor is requesting more student preparation for working online.
- Students are also lacking in problem solving ability. One instructor recommends removing the 'be familiars' from the course outcomes.
- Several instructors requested a common final exam in COP2210 or an entrance exam to COP3337. A common theme is that the outcomes for COP2210 must be met before students can progress to COP3337.

COP 3530 Data Structures

- Students agreed with the overall Valuation of Outcomes as well as their Coverage moderately (4.55 and 4.42 respectively).
- There is no significant concern about the outcomes expressed in the Students Suggestions section.
- From instructor course appraisals, students seem to be deficient in linked lists, stacks, collections and recursion.

COP 4226 Advanced Windows Programming

- Students agreed with the overall Valuation of Outcomes as well as their Coverage moderately (4.49 and 4.52 respectively).
- One of the outcomes for the course includes database connectivity. A database course is not a prerequisite for this course, so it is difficult to cover database connectivity adequately. The instructor recommends removing database connectivity from the outcomes.

COP 4338 Computer Programming III

- Students agreed with the overall Valuation of Outcomes as well as their Coverage moderately (4.58 and 4.09 respectively).
- Students complained about the presentation of the material by an instructor.
- From instructor course appraisals, students seem to be deficient in problem solving and documentation standards.
- Instructors would like more time to be able to cover multi-threading and synchronization. It would be beneficial if students already knew UNIX before this course.
- One instructor noted that students were deficient in pointers and C data structures. We do not have a prerequisite course that could cover pointers, C data structures, or UNIX.

COP 4520 Introduction to Parallel Computing

- Students agreed with the overall Valuation of Outcomes as well as their Coverage strongly (4.83 and 4.62 respectively).

Subject Area: Software Engineering (SAC: Masoud Sadjadi)

CEN 4010 Software Engineering I

- Students agreed with the overall Valuation of Outcomes strongly (4.78) and with their Coverage, strongly/moderately (4.46).
- **Instructors' comments (as reported by the SAC):**
 - ❖ The course objectives should be evaluated to provide a more modern approach to software development. Some of the concepts which are covered rely on waterfall development which is very hard to find in practice under most modern product development shops.
 - ❖ As the professor of this course, I have no objections to the current listed pre-requisites. With that being said, I do hear grumblings from the students that clearly indicate that they are split on the necessity of Net-Centric for this course. It might be worth having a discussion about the knowledge set required to be successful in CEN4010. I have found in the two semesters that I have taught this course, that while students complain about their perceived preparedness for the course, they typically find a way to have a working finished product at the completion of the course.
 - ❖ Since the expectation is that students know the Agile software development process prior to Senior Project, I do believe it is time to evaluate the text for this course. Currently the text presents the waterfall method and as such the professor is sort of bound to this method. This semester, I presented both methods and allowed the student-teams to decide which method they wanted to leverage for their product development. This typically leaves half the class underexposed to this method going into the Senior Design project. It is my recommendation that we seek a text that better aligns with the expectations of the follow-on course in order to better prepare students for that capstone course.
 - ❖ Students are generally prepared technically but struggle immensely with navigating teamwork. Opportunities in prior courses to work in teams might aid in developing skills for navigating challenges associated with working with others.
 - ❖ Given that this course is the prerequisite to Senior Design, I believe that an update to the text to align with the expectations of the follow-on course would serve the students better. The current text adopts and advocates for the waterfall process (which the students should be made aware of); however, it might serve the students better to adopt a text that better aligns with expectations - an agile methods book.
- **Students' comments:**
 - ❖ Prepare students more on how to work in a team efficiently (Code Sharing, git, etc.)
 - ❖ My only complaint was that the class was held in the evening, and because it is largely based on a group project, my teammates and I often felt lethargic by the time we attended.
 - ❖ This course is straight forward and handles group dynamics very well. In hindsight, I would suggest making Net-Centric Computing a pre-requisite for this course. Many students come into this course with no knowledge of simple application functions such as GET and POST request. I will additionally mention that is no required CS course dealing with front end manipulation so I would suggest including it with some part of a class.

- ❖ We need actual software development and less paperwork.
- ❖ Don't calculate points toward our grade for participation. FIU is a commuter school, and as such, MANY of us have to drive 30-45 minutes, in GOOD conditions (depends greatly since we're in Miami), and some drive even LONGER in good conditions. Sometimes, it's hard to get there on time when you live so far. Sometimes it's hard to get there at all due to a classic Miami traffic jam (we all know how long those can last). So, don't penalize us for participation... If we feel we can learn the material for that day on our own, let us do that please.
- ❖ I do not understand why Net-centric is a Co-requisite. Both classes have nothing in common so far. The only way I can see they can relate is if Net-centric should have a project that can be done using software testing.
- ❖ I would love to have this class be thought without having to take other courses. That way it will be closer to the real work experience.
- ❖ Split into two courses, one about planning and introduction to application stacks (with homework to learn front-end and back-end frameworks) and the second part revolving around creating an application
- ❖ One of the greatest classes I have ever taken. More classes should be taught with the openness that the instructor teaches.
- ❖ This class is rather well formatted already. I think slightly more emphasis should be put on making a functioning program, but the class felt very smooth as is already.
- ❖ The weekly quizzes on material being discussed in class and only to key concepts of the weekly readings. Since all the assignments are group related this made sure every student did the weekly readings which kept my team prepared every week for the new material ahead.
- ❖ The emphasis on UML modeling is useless. The required text is useless, waterfall is not as widely used as before. Real companies are adopting agile teams.
- ❖ Git MUST be introduced BEFORE this course. The department is failing its students to not mention it before this course.
- ❖ The work given throughout the course was okay. The readings really helped us get to know different points and topics that affect software development and deployment like algorithmic accountability.
- ❖ It is a very valuable perspective for a current industry professional to come in and share practical experience about the software development cycle. More focus on practical knowledge could improve the program.
- ❖ The course failed in about every outcome. First, the course lacked lectures. No lecture ever lasted more than 15 minutes and after the second half of the course lectures were dropped altogether. After that class time was set up for teams to work on their project. But 90% the grade could be archive by writing a paper, so many students passed without writing a single line of code. Be familiar with the Software Development Life Cycle: I do not know Software Development Life Cycle stands for. Master the techniques to gather and specify the requirements of a medium-size software system using UML: The UML lecture did not last more than 15 minutes or about 2 slides. Then I was given a document which seemed to be taken from a Google Search. I don't know how to do UML. Master the

techniques to design and implement a medium-size software system: I don't think it is possible to learn about how to implement medium-size software system if student could pass the class without writing code. Be familiar with software testing techniques: No resource about testing was ever shared. I was given a chart a "sample test cases". And I inferred what was testing was from that. If there was a lecture about this it did not last more than 15 minutes, and I can't recall those 15 minutes of my life. Be familiar with system walkthroughs: Never talk about nor asked about this. Be familiar with software documentation: Never talk about nor asked about this. Demonstrate the ability to communicate the details of the technical solution through verbal and written modes: Student were asked to do a presentation about their project, but it was not a technical presentation, not code of the system was really shown. It was more of a product showcase to a nontechnical audience. Moreover, for students who did not have a project could ramble about "Introduce the team including roles and responsibilities," "description of the customer/setting for the project," "Salient characteristics of the customer" etc. This downgraded the presentation to ENC 3213 presentation instead of CEN 4010 presentation. All in all, this felt like a technical writing class not a software engineering class. Suggestions: -Clear and objective descriptions of requirements and expectations. -Student should be graded on how well they can complete a project, the paper and presentation should be complementing not the main thing.

CEN 4021 Software Engineering II

- Students agreed with the overall Valuation of Outcomes as well as their Coverage strongly (4.87 and 4.90 respectively).
- **Instructors' comments:**
 - ❖ The students were lacking knowledge in the area of modeling software artifacts using UML. The students lack the ability to create both static and dynamic UML models. They were also not proficient in the use of any UML modeling tool.
 - ❖ More coverage on software design and software architecture.
 - ❖ Deeper study in the SDLC
- **Students' comments:**
 - ❖ More guidance/specifics on what's required for the Deliverables would help.
 - ❖ The class was very interesting and exposed the students to the software engineering process very well. However, the preparation for this course was nothing compared to the amount of work required from the class. It would have been better if Software Engineering 1 would have prepared the students better for this course in terms of UML use.
 - ❖ This course has helped us tremendously by showing us the way the Software Engineering Industry works. All the different Panels were very important for learning from important aspect of Software Development from Software Architecture to Project Management. I'm very grateful I took this course and I believe it had a great impact in my professional life.

CEN 4072 Software Testing

- Students agreed with the overall Valuation of Outcomes as well as their Coverage moderately (4.58 and 4.21 respectively).
- **Instructors' comments:**
 - ❖ Students are lacking knowledge of some mathematical concepts that helps with test generation. For example, relations and equivalent classes.
 - ❖ Students should be introduced to the concept of a server and manipulating the actions of the server.
 - ❖ It is good to see that the Objective 6 - "Be exposed to program debugging", has been removed.
 - ❖ Students lack some basic problem-solving skills such as drawing a flowchart for a single method and tracing the values passed to the method. This is a necessary skill for performing program inspections and code coverage.
 - ❖ Students should be exposed to working in teams and team management before taking this course. Assuming this is possible with the curriculum.
 - ❖ Some students expect to be spoon-fed and are not willing to use the wide array of resources available to learn how to use the various testing tools. In addition, some students wait until the last minute to start a project that is way too complex to complete in one or two days.
- **Students' comments:**
 - ❖ Use new tools used by more companies such as selenium instead of RFT.
 - ❖ Some examples on how to use testing tools would be nice. Online resources were not helpful.
 - ❖ The board work was useful for teaching the written problems for this course. More hands-on experience with the testing tools would have been worthwhile, rather than letting it be free range. Summary of important material was handled well.
 - ❖ I understand that we are this late into our major and that we should be able to figure how things work. However, it would be good if the usage of the actual tools is taught instead of teaching some of the theoretical concepts of software testing. It would be easier to do the actual testing.
 - ❖ There need to be more recourses for setting the testing software up given at the beginning of the course.
 - ❖ This course would be no less effectively if it did not require a textbook.
 - ❖ Suggest students not to take it earlier or after software engineering. Many concepts are needed that build into for this course
 - ❖ To improve, there should be formal tutorials that address the possible problem one may encounter while trying to set up IBM RFT and Cobertura. From my experience with the class, one can easily waste 50 to 80 hours trying to set up that 2 software. Imagine how much efficient a student would be in testing and getting code coverage if he/she didn't have to waste so much time on those. That's why those tutorials should be considered since they are most needed resources.
 - ❖ Please consider recording classes, for those that miss it, in order to catch up.
 - ❖ Class had some components that were never taught in previous classes.

CIS 4911 Senior Project (also IDS 4918)

- Students agreed with the overall Valuation of Outcomes as well as their Coverage moderately (4.69 and 4.26 respectively).
- **Instructors' comments:**
 - ❖ CEN 4010 should include Agile/Scrum software development in its syllabus to better prepare students for this course.
- **Students' comments:**
 - ❖ Perhaps try to get one or two more sprints in the summer.
 - ❖ It was fine, just need a way to help when teammates drop course.
 - ❖ It is a bit confusing when starting the project; therefore, I recommend having clearer instructions on what to do at the beginning of the senior project semester.
 - ❖ Make Product Owners formally agree to be available as per our scheduled work times, there were cases where product owners were not always available for Sprint Review meetings or Planning.
 - ❖ More guidance during the process would have been very helpful. I think we did not have a mentor that would have fulfilled the role.
 - ❖ Most projects focused on web development. It would have been nice to have been given a heads-up earlier in my academic career. Also, for students like me who provide for their own living the work and school life balance are rough.
 - ❖ Please let students pick their own project ideas. You can have certain minimum requirements that they must meet, but it would be great if they idea was theirs. I had a great idea for an application I wanted to do for senior, but I did not get the chance because of the current way things are.
 - ❖ Only real complaint was when asking about UML diagrams / documentation, was told that I should have "learned that already". I HAD learned about UML already, but a lot of the rules are poorly defined, and, in my experience, different graders have different preferences for what is "correct".
 - ❖ I think that the previous student, should left comment on the different thing to change, and the new feature that could be a good thing to work on.
 - ❖ Should focus agile in software engineering and provide some web development classes
 - ❖ I wish it was more structured, but it's a senior capstone course so I guess I can't really ask for that.
 - ❖ Needed more time. Lost a week in the beginning of an already short semester.
 - ❖ Give a little more information background on what the project is that we are working on. The old resources were very difficult to find.
 - ❖ Better communication of expectations at the beginning of the semester (i.e. documentation).
 - ❖ More mandatory contact with instructor.
 - ❖ Less documentation. In industry, class diagrams are sequence diagrams are barely ever used.
 - ❖ Information needs to be organized for this course; everything needs to be in one place. Some info is on Moodle, some is on the schedule, and some is on google drive. There is no reason why you have to check 3 or 4 different locations to be find the complete instructions for a single deliverable.
 - ❖ I believe what this course needs are mandatory weekly meetings were us students can receive meaningful input from both the professors and other fellow students

so that it isn't an all on your own type of class, where the only communication that occurs is over email.

- ❖ No feedback was given besides "looks good".
- ❖ Not very organized, conflicting documentation, no knowledge of class performance throughout the semester. Need to give students a week to look at the project list to allow them to thoroughly look through the projects they desire to join.
- ❖ It was a struggle to figure out what was due when and with what requirements. The professor demanded strict adherence to his instructions which were often unclear or conflicting with things posted online. This class would benefit greatly from a calendar with all requirements posted accurately.
- ❖ Get a project management system that works. The servers went down at least once per sprint for 1-2 days at a time. One instance caused all teams to lose days' worth of work. Documentation requirements were vague and were often amended 1 day before deadlines.
- ❖ While the execution of this type of course is essential to a student's ability to fully grasp the software engineering process, I do feel there is a great deal of disconnect between the courses we are required to take prior and the practical application of that knowledge.
- ❖ I strongly believe there could be other courses on the computer science curriculum that can benefit the preparation and skills needed for the senior project. For example, a course that goes in depth on client and server-side applications, maybe some projects that simulate scalability, etc... But either way, I enjoyed the senior project class. The large amount of documentation is a bit excessive in my opinion but all together I find the class great.
- ❖ This was the first semester that we used JIRA. There was a learning curve at the beginning but became easier to use as the semester progressed. Confluence was down a couple of times when deliverables were due, I assume it was because everyone was trying to use it at the same time. So, that should be addressed moving forward. Also, if JIRA is to be used in future semesters it would be beneficial if students had the opportunity to use it in Software Engineering Course beforehand. That way the transition is seamless into Senior Project. Finally, in the beginning of the semester when projects are picked, it would be nice if the product owners did a presentation for the entire class showcasing the previous semesters work and what they want moving forward. That way students are a bit more informed when picking a project for the rest of the semester.
- ❖ The material was sparsely relevant to my career and the relevant components were rehashed straight from the Software Engineering course we're required to take right before this class. 8) I think the organization of documentation and using jira/confluence was very confusing. There was differing information.
- ❖ understand the class is about Agile development and adapting to change, but as someone who has worked in the field for the last two years, I actually do not use most of the tactics this course is supposed to teach us. In fact, I found it quite cumbersome. There should be less of a focus on the documentation and more on actual development. I understand students can develop bad habits with their first introduction to full-stack development, but it is the only course offered by FIU

(aside from CEN4072 - Fundamentals of Software testing) that actually gives us a look into real life scenarios and delivering a tangible artifact to the customer.

- ❖ We don't get enough preparation in the career to work in the final project. If you don't work in a real project, then you don't have enough preparation to work in a project like the ones in the Senior Project
- ❖ Course is the best experience in college. I work at an enterprise level and there is nothing like what areal software engineer job is than this course.

C. Graduating Student (Exit) Survey of Student Outcomes

The Student Outcomes Survey is completed by students in the semester in which they expect to graduate. Each student rates each outcome with respect to two criteria, attainment and relevance.

Attainment: *This program outcome has been met for me personally*

5: *I agree strongly*
4: *I agree moderately*
3: *I agree somewhat*

2: *I disagree somewhat*
1: *I disagree moderately*
0: *I disagree strongly*

Relevance: *How meaningful do you consider this outcome to be for you personally?*

5: *Extremely meaningful*
4: *Moderately meaningful*
3: *Somewhat meaningful*

2: *Somewhat meaningless*
1: *Moderately meaningless*
0: *Extremely meaningless*

Please note that as we could not change this survey until Fall 2017 (after the last ABET six-year-Evaluation year, 2016-2017), the data collected in Summer 2017 is for the questionnaire matching the pre-Fall 2015 Student Outcomes. The mapping matrix between the two sets of Student Outcomes is given below as Figure-3. For all other semesters from Fall 2017 through Spring 2019, the data collected matches the current Student Outcomes.

Figure-3:

Mapping between Pre-Fall 2015 and Beginning-in-Fall-2015 Student Outcomes

[For this evaluation period, the mapping is used solely for Summer 2017 data.]

Mapping Between Pre-Fall-2015 and Beginning-in-Fall-2015 Student Outcomes								
Beginning-in-Fall-2015 Student Outcomes	Pre-Fall 2015 Student Outcomes							
	Outcome	Outcome	Outcome	Outcome	Outcome	Outcome	Outcome	Outcome
	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
								Pre-Fall-2015 Student Outcomes
a) Ability to apply knowledge of Computing and Mathematics	X							a) Demonstrate proficiency in the foundation areas of Computer Science - discrete structures, logic, algorithms.
b) Ability to analyze a problem, and define its computing requirements		X	X					b) Demonstrate proficiency in various areas of CS including data structures, programming languages and computer systems.
c) Ability to design, implement, and evaluate a computer-based system	X	X	X	X				c) Demonstrate proficiency in problem solving and application of software engineering techniques.
d) Ability to function effectively on teams						X		d) Demonstrate mastery of at least one modern programming language and proficiency in at least one other.
e) Understanding of professional, ethical, legal, security, and social issues					X			e) Demonstrate understanding of the social and ethical concerns of the practicing computer scientist.
f) Ability to communicate effectively							X	f) Demonstrate the ability to work cooperatively in teams.
g) Ability to analyze local and global impact of computing					X			g) Demonstrate effective communication skills.
h) Recognizing the need to engage in continuing professional development					X		X	h) Have experience with contemporary environments and tools necessary for the practice of computing
i) Ability to use current techniques, skills, and tools necessary for computing practice		X	X				X	
j) Ability to apply mathematical foundations of computing in designing computer-based systems	X		X					
k) Ability to apply software engineeringt principles to develop software			X					

Data was collected (number of responses is in parenthesis) for Summer 2017 (25), Fall 2017 (48), Spring 2018 (21), Fall 2018 (8), and Spring 2019 (8) for a total of 110 responses during the period of this Assessment. Raw data and calculation of statistics for each semester is presented in Appendix D-1. The summary of the whole is presented in Appendix D-2.

The following table summarizes the responses of 110 graduating students completing the survey between summer 2017 and spring 2019. The mean responses are expressed as percentages of 5, the maximum rating.

<u>Exit Survey (Graduating Students) 110 Respondents</u> <u>Student Outcomes</u>	<u>Outcome Attainment</u>		<u>Perceived Relevance</u>	
	<u>Mean</u>	<u>Percentage</u>	<u>Mean</u>	<u>Percentage</u>
a: Ability to apply knowledge of Computing and Mathematics	4.53	90.6	4.64	92.8
b: Ability to analyze problem – identify and define its computing requirements	4.56	91.2	4.87	97.4
c: Ability to design, implement, and evaluate a computer-based system	4.37	87.4	4.82	96.4
d: Ability to function effectively on teams to accomplish a common goal	4.54	90.8	4.77	95.4
e: Understanding of professional, ethical, legal, security, and social issues	4.35	87.0	4.37	87.4
f: Ability to communicate effectively with a range of audiences	4.51	90.2	4.72	97.4
g: Ability to analyze local and global impact of computing on society	4.31	86.2	4.35	87.0
h: Recognition for the need for and an ability to engage in continuing professional development	4.43	88.6	4.75	95.0
i: Ability to use current techniques skills, and tools necessary for computing practice	4.19	83.8	4.78	95.6
j: Ability to apply mathematical foundations and algorithmic principles in design of computer systems	4.47	89.4	4.63	92.6
k: Ability to apply design and development principles to construct complex software systems	4.46	89.2	4.78	95.6
	====	====	====	====
Average Ratings of Student Outcomes	4.43	88.6	4.68	93.6
	====	====	====	====
Overall Satisfaction for CS Areas, Outcomes ‘a’, ‘b’, ‘c’, ‘e’, ‘g’, ‘i’, ‘j’, and ‘k’:	4.40	88.0	4.66	93.2

Table 3: Exit Survey of Attainment & Relevance of Student Outcomes

The Average Rating Scores of Student Outcomes with respect to attainment (4.43) and perceived relevance (4.68) are slightly higher than those found in the previous Assessment cycle (4.34 and 4.60 respectively).

D. Alumni Survey of Program Educational Objectives

Alumni responding to the survey are asked to rate the contribution of their broad educational experience at FIU to their personal growth, capacity for life-long learning, communication skills,

social and ethical awareness, career preparation, and preparation for graduate study. They rate their preparation in the major areas of the BS-CS curriculum. The respondents also provide “overall” ratings of their FIU educational experience and the student’s preparation at graduation. Finally, the alumni provide a rating of their overall satisfaction with the BS in CS program.

Responses to the survey questions are on a the following scale

4: Excellent, 3: Good, 2: Satisfactory, 1: Poor and 0: Unsatisfactory

Please note that as we could not change this survey until Fall 2017 (after the last ABET six-year-Evaluation year, 2016-2017), the data collected is for the questionnaire matching the pre-Fall 2015 Program Objectives. The mapping matrix between the two sets of Student Outcomes is given below as Figure-4.

Figure-4:

Mapping between Pre-Fall 2015 and Beginning-in-Fall-2015 Program Objectives

[For this evaluation period, the mapping is used solely for Summer 2017 data.]

Pre-Fall 2015 Program Objectives	Beginning in Fall 2015 - Program Objectives					Beginning in Fall 2015 - Program Objectives
	1	2.1	2.2	2.3	2.4	
1. To provide our graduates with a broad-based education that will form the basis for personal growth and life-long learning.					X	1. Be successful in applying for entry level professional positions in computing-related fields, or for admission to graduate programs.
2. To provide our graduates with a quality technical education that will equip them for productive careers in the field of Computer Science.		X				2. Be prepared for career accomplishment, responsibility and advancement in computing-related professions by virtue of having received
3. To provide our graduates with the communication skills and social and ethical awareness requisite for the effective and responsible practice of their professions.			X	X		2.1 A high-quality technical education in computing,
4. To prepare students for BS level careers or continued graduate education.	X					2.2. Communication and team-work skills,
						2.3 Awareness of the ethical and social responsibilities of their profession,
						2.4 An ability to engage in continued professional development activities.

The table below summarizes the responses to this survey. The means for the current survey cycle, May 2019 to Nov 2019, are compared with corresponding means for earlier cycle, May 2017 to Oct. 2017. The numbers in the first column refer to the BS-CS Program Objectives included in Appendix A-1. The raw data for the current cycle along with the statistical results for the current assessment period are presented in Appendix E-1.

<u>Alumni Survey of Program Objectives</u>		May 2019 to Nov. 2019		May 2017 to Oct. 2017	
		122 Respondents		211 Respondents	
<u>Program Educational Objective</u>		<u>Outcome Attainment</u>		<u>Outcome Attainment</u>	
		<u>Average</u>	<u>Percentage</u>	<u>Average</u>	<u>Percentage</u>
2.4	Capacity for personal growth	3.39	84.72	3.25	81.25
2.4	Capacity for life-long learning	3.39	84.72	3.31	82.75
2.2	Development of communication skills	3.11	77.78	3.04	76.00
2.3	Awareness of social, ethical responsibility	3.06	76.39	2.95	73.75
1	Preparation for career in CS	3.11	77.78	2.94	73.50
1	Preparation for graduate study	2.92	72.92	2.95	73.75
	Overall preparation upon graduation	3.06	76.56	3.06	76.50
2.1	Computer Programming	3.36	84.03	3.16	79.00
2.1	Systems Development	2.78	69.44	2.89	72.25
2.1	Data Structures & Algorithms	3.25	81.25	3.21	80.25
2.1	Computer Architecture & Organization	2.86	71.53	2.96	74.00
	Overall FIU educational experience	3.16	79.05	3.07	76.75
	Overall satisfaction with BS-CS program	3.12	78.05	3.10	77.50

Table 4: Alumni Survey of Attainment of Program Educational Objectives

E. Employer Survey of Program Educational Objectives

This is the third biennial Assessment for which we have surveyed the Employers of our students and the members of the Industrial Advisory Board of the School (many employ our graduates). The survey instrument is included in Appendix E-2. The raw data along with statistical results is included in Appendix E-3, and the results are included in the table below. Note that the participation for this survey is pretty low (9 responses; only 5 completed).

<u>Employer Survey of Program Objectives</u>		<u>(9-4) Respondents</u>	
		<u>Outcome Attainment</u>	
<u>Program Educational Objective</u>		<u>Average</u>	<u>Percentage</u>
2.4	Ability to learn new Emerging Concepts	3.60	90.00
2.1	Mastery of CS concepts & ability to solve problems	3.40	85.00
2.2	Ability to communicate verbally	3.80	95.00
2.2	Ability to communicate in written form	3.80	95.00
2.3	Understanding of social, ethical concerns	3.50	87.50
2.2	Ability to work cooperatively in a team	3.80	95.00
1	(Will you consider hiring our graduates – 5-YES, 0-NO) (an important gauge of Preparation for career in CS)	4.00	100.00
OVERALL ATTAINMENT OF PROGRAM OBJECTIVES		3.66	91.50

Table 5: Employer Survey of Attainment of Program Educational Objectives

F. Course Embedded Direct Assessment

SCIS began applying course-embedded assessment of the BS in CS program in fall 2010 in order to supplement the direct measures obtained via capstone assessment in the Senior Project (see the following section). This strategy was applied using either multiple-choice (M-C) quizzes or observing student assignments and/or projects. Appendix-F contains the Direct Assessment Summaries for all courses subject to this direct assessment excluding the Senior Project. Most of the student ratings are based on their performance in M-C quizzes and a few observations are derived from their assignment work.

The evaluation of these assessments is included in section IV.B (Evaluation – Student Outcomes).

G. Capstone Project Assessment

Current requirements of the BS in Computer Science include completion of a capstone course, CIS 4911, Senior Project. Beginning with the first offering of CIS 4911, SCIS has performed assessment of all Student Outcomes via evaluation of the presentations and artifacts of all completed projects. Each project is rated by 2 or more evaluators (exactly 2 in this assessment cycle) according to a rubric *Senior Project Assessment of Student Outcomes of the BS in Computer Science*, and scored on the following scale:

Rating	Criterion
n/a	The project does not provide clear evidence about this particular outcome
1	The project demonstrates poor attainment of this outcome
2	The project demonstrates fair attainment of this outcome
3	The project demonstrates good attainment of this outcome
4	The project demonstrates very good attainment of this outcome
5	The project demonstrates excellent attainment of this outcome

The current version of the rubric was finalized in spring 2015, and these Direct Measurements apply to the Student Outcomes effective in Fall 2015. The rubric and associated check-list and score grid are included as Appendix G-1 of this report.

The data from these semester-wise assessment events are summarized in Appendices G-2 through G-7. The summary evaluation of these assessments is included in Appendix G-8, and is presented in Section IV.B (Evaluation – Student Outcomes).

IV. EVALUATION

In this section of the report, the data presented in the previous section are evaluated. For quantitative data, the threshold value at which SCIS deems a measured item to satisfy its criteria is 75% of the maximum attainable rating.

<u>Measured Item</u>	<u>Scale</u>	<u>Threshold</u>
Course Outcomes	1 to 5	3.75
Student Outcomes	0 to 5	3.75
Program Objectives	0 to 4	3.00

A. Course Outcomes

The Subject Area Coordinator (SAC) reports (Appendix C) present the data obtained for each course via surveys by students and instructors. The Course Outcomes for each required or elective course of the BS in Computer Science program are evaluated for relevance and attainment by the SAC. Their evaluations are contained in the SAC reports.

The evaluation of the Course Outcomes by the Assessment Coordinator (AC) is based on the student ratings of the course outcomes summarized in Table 1.

AC-Evaluation-01: The data for Course Outcomes by Student Surveys for COP 4604 (Advanced UNIX Programming and COT 4521 (Introduction to Computational Geometry – each taught only once during the period of evaluation) and the MAD courses (2104 and 3512) are not available. Note that SCIS began offering its own Discrete Mathematics course (COT 3100) in Spring 2017 to replace MAD 2104. The course outcomes data for this course are very impressive.

AC-Evaluation-02: The Value of Course Outcomes rating of every course for which data are available, exceeds the 3.75 acceptability threshold. In fact, students ascribe at least **high** value (4.00 or higher) to the outcomes of every course except two (CAP 4630 and CEN 4083) with the rating of the Value of Course Outcomes of over 86% of courses (25 out of 29) is **very high** (4.40 or higher). The overall rating for the Value of Course Outcomes is 4.59 which is essentially the same as observed in the last two Assessment cycles (4.63 and 4.59).

AC-Evaluation-03: The student rating of the Adequacy of Coverage of Course Outcomes for every course except two (CAP 4630 - 3.38, and CEN 4083 – 3.25) exceeds the acceptability threshold of 3.75. In fact, students ascribe at least **high** value (4.00 or higher) to the adequacy of coverage of all except two courses (CDA 4625 and COP 4722), with the rating of **very high** (4.40 or higher) for fourteen out of 29 courses. The overall rating for the Adequacy of Coverage of Course Outcomes is 4.34 which is just a bit lower than that observed in the last two Assessment cycles (4.52 and 4.42).

AC-Evaluation-04: Note that in this assessment cycle, the overall student participation is lower than in the last cycle. This may be due to the migration of the evaluation process to fully online mode after Spring 2018. SCIS discontinued the practice of taking the netbook computers in classes

to force the students to complete the surveys. We should explore student incentives (priority in advising, student workshop registrations, etc.) to entice more students to complete these surveys.

B. Student Outcomes

Evaluation of the level of attainment of the BS in CS Student Outcomes utilizes data obtained via several direct and indirect assessment mechanisms listed below:

Indirect Mechanisms:

- The Graduating Student (Exit) Survey,
- Course Outcomes Surveys by Students and by Instructors.

Direct Mechanisms:

- Capstone Project Assessment via CIS 4911 Senior Project presentations,
- Course-embedded Assessment by multiple-choice questions in several required courses taken by the BS-CS majors: COT 3100 (Discrete Structures – offered by SCIS starting in Spring 2017), MAD 2104 (Discrete Mathematics), MAD 3512 (Theory of Algorithms), COP 3337 (Programming II), COP 4338 (Programming III), COP 3530 (Data Structures), COP 4710 (Database Management), COP 4555 (Principles of Programming Languages), COP 4610 (Operating Systems), and CEN 4010 (Software Engineering I).
- Course-embedded Assessment by portfolio inspection in CGS 3095 (Ethics and Social Issues in Computing).

The direct assessment events performed from summer 2017 to spring 2019 are documented in the summary provided in Appendix F. The rating sheet and the rubric used for evaluation of Senior Project for assessment of Student Outcomes is provided in Appendix G-1.

a) An ability to apply knowledge of computing and mathematics appropriate to the program’s student outcomes and to the discipline.

Indicators

1. Graduating Student Ratings	Relevance 92.71%	Attainment 90.54%	Sample: 110
2. Course Outcomes CAP 4630	Value: 75.8%	Coverage: 67.6%	Sample: 24
3. Course Outcomes CAP 4770	Value: 98.0%	Coverage: 91.8%	Sample: 36
4. Course Outcomes COP 4520	Value: 96.6%	Coverage: 92.4%	Sample: 7
5. Course Outcomes COP 4534	Value: 92.8%	Coverage: 84.0%	Sample: 20
6. Course Outcomes COT 3100	Value: 86.4%	Coverage: 86.8%	Sample: 78

- 7. Course Outcomes COT 3541 Value: 95.0% Coverage: 96.0% Sample: 42
- 8. Course Outcomes MAD 2104 Data Not Available [Substituted by COT 3100]
- 9. Course Outcomes MAD 3512 Data Not Available
- 10. Course-Embedded Assessment – COT 3100

Fall 2017 Event: 29 students completed a 24-question multiple choice quiz.
Criterion: At least 75% of students should score 75% or higher.
Observation: 22 out of 29 (75.86%) students scored at least 18 points.
Summary Observation: Clearly, since we started teaching our own Discrete Structures course, the performance of students has substantially improved (see below for MAD 2104).

- 11. Course-Embedded Assessment - MAD 2104

Fall 2017 Event: 36 students completed a 16-question multiple choice quiz.
Criterion: At least 75% of students should score 75% or higher.
Observation: 20 out of 36 (55.56%) students scored at least 12 points.
Summary Observation: 20 out of 36 (55.56%) students demonstrated proficiency in Discrete Mathematics. This is about the same as determined in the previous assessment.

- 12. Course-Embedded Assessment - MAD 3512

Spring 2018 Event: 15 students completed a 10-question multiple choice quiz.
Criterion: At least 75% of students should score 75% or higher.
Observation: 7 out of 33 (21.21%) students scored at least 7.5 points. This is substantially worse than 60.0% we observed in the last assessment. 12 out of 33 (36.36%) students scored at least 7 points (70% or higher).
Summary Observation: 7 out of 33 (21.21%) students demonstrated proficiency in Theory of Algorithms. The result in the previous assessment was substantially higher, equal to 60.0%.

- 13. Senior Project Assessment

Event: Artifacts of all completed Senior Projects are assessed, by application of the *Senior Project Assessment of Student Outcomes of the BS in Computer Science* rubric, for attainment of outcome a). This event was replicated in all semesters from summer 2017 to spring 2019.
Criterion: Attainment should be rated at **75% or 3.75** on a 1—5 scale, or better.
Observation: Summer 2017: 3.06 Fall 2017: 3.06 Spring 2018: 3.08
Summer 2018: 3.08 Fall 2018: 3.08 Spring 2019: 3.14
Weighted over the entire assessment period (106 projects): 3.09

Outcome Evaluation: Graduating students consider this Student Outcome highly relevant, and almost 93% believe that they have attained it. Indicator 2 exceeds the acceptable threshold for the Value of Outcomes, but falls slightly below in the Coverage. Indicators 3 through 7

substantially exceed the acceptable threshold for the Value and the Coverage of Course Outcomes for CAP 4770, COP 4520, COP 4534, COT 3100, and COT 3541. Indicator 10 clearly shows the improvement in student performance since we started teaching Discrete Mathematics in SCIS. Indicators 11 and 12 clearly show that students do not attain the desired acceptable level of proficiency for MAD 2104 and MAD 3512. For MAD 2104, the performance is similar to the one detected in the previous assessment cycle, while it is substantially lower for MAD 3512. Mad 2104 is now substituted by our own course, COT 3100, and we clearly see an improvement in both, the Value and Coverage of Course Outcomes. Finally, indicator 12 shows that our Senior Projects have not quite incorporated this curriculum component to a significant level. Attainment of Student Outcome (a) is rated as **barely acceptable**.

b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

Indicators

1. Graduating Student Ratings	Relevance 97.31%	Attainment 91.23%	Sample: 110
2. Course Outcomes CAP 4104	Value: 98.0%	Coverage: 96.0%	Sample: 58
3. Course Outcomes CAP 4630	Value: 75.8%	Coverage: 67.6%	Sample: 24
4. Course Outcomes CAP 4641	Value: 98.4%	Coverage: 95.6%	Sample: 36
5. Course Outcomes CAP 4710	Value: 97.6%	Coverage: 82.4%	Sample: 1
6. Course Outcomes CAP 4770	Value: 98.0%	Coverage: 91.8%	Sample: 36
7. Course Outcomes CDA 3103	Value: 84.8%	Coverage: 80.0%	Sample: 82
8. Course Outcomes CDA 4101	Value: 88.0%	Coverage: 83.6%	Sample: 64
9. Course Outcomes CDA 4625	Value: 95.8%	Coverage: 78.4%	Sample: 24
10. Course Outcomes CEN 4010	Value: 95.6%	Coverage: 89.2%	Sample: 58
11. Course Outcomes CEN 4021	Value: 97.4%	Coverage: 98.0%	Sample: 17
12. Course Outcomes CEN 4083	Value: 77.6%	Coverage: 65.0%	Sample: 2
13. Course Outcomes COP 3530	Value: 91.0%	Coverage: 88.4%	Sample: 97
14. Course Outcomes COP 4338	Value: 91.6%	Coverage: 81.8%	Sample: 96
15. Course Outcomes COP 4555	Value: 88.8%	Coverage: 87.8%	Sample: 45

- | | | | |
|---|---------------|-----------------|------------|
| 16. Course Outcomes COP 4610 | Value: 93.8% | Coverage: 88.8% | Sample: 90 |
| 17. Course Outcomes COP 4710 | Value: 95.20% | Coverage: 89.0% | Sample: 75 |
| 18. Course Outcomes COP 4722 | Value: 89.0% | Coverage: 79.0% | Sample: 42 |
| 19. Course-Embedded Assessment – CEN 4010 | | | |

Spring 2018 Event: 24 students were evaluated for their Project Documentation.

Criterion: 75% of students should score at least 75% or higher.

Observation: 13 out of 24 (54.17%) students answered at least 8 questions correctly. 16 out of 24 students answered at least 7 questions correctly.

20. Course-Embedded Assessment - COP 4555

Fall 2017 Event: 32 students completed a 10-question multiple choice assessment quiz.

Criterion: 75% of students should score at least 75% or higher.

Observation: 46.88% of the students answered at least 8 questions correctly. 23 out of 32 (71.88%) students scored at least 7 points (70% or higher).

21. Course-Embedded Assessment - COP 3530

Fall 2017 Event: 33 students completed a 16-question multiple choice assessment quiz.

Criterion: 75% of students should score at least 75% or higher

Observation: 23 out of 33 (69.7%) students answered at least 12 questions correctly.

22. Course-Embedded Assessment - COP 4710

Spring 2018 Event: 54 students completed a 16-question multiple choice assessment quiz.

Criterion: 75% of students should score at least 75% (12) or higher.

Observation: 20.37% of the students (11 out of 54) answered at least 12 questions correctly. 31 out of 54 (57.41%) of the students answered at least 10 questions correctly.

23. Course-Embedded Assessment – COP 4338 (Computer Systems – Processes)

Fall 2017 Event: 14 students completed a multiple choice assessment quiz worth 8 points.

Criterion: 75% of students should score at least 6 points.

Observation: 78.57% of the students answered at least 6 questions correctly.

24. Course-Embedded Assessment – COP 4610 (Systems – Storage Management)

Fall 2017 Event: The artifacts (submitted programs/projects) of 35 students were evaluated against the appropriate rubrics with the maximum possible score being 12.

Criterion: 75% of students should score at least 9 points.

Observation: 60.0% of the students scored at least 9 points.

25. Course-Embedded Assessment – COP 4610 (Systems – Memory Management)

Fall 2017 Event: The artifacts (submitted programs/projects) of 39 students were evaluated against the appropriate rubrics with the maximum possible score being 12.

Criterion: 75% of students should score at least 9 points.

Observation: **87.18%** of the students scored at least 9 points.

26. Senior Project Assessment

Event: Artifacts of all completed Senior Projects are assessed, by application of the *Senior Project Assessment of Student Outcomes of the BS in Computer Science* rubric, for attainment of outcome b). This event was replicated in all semesters from summer 2017 to spring 2019.

Criterion: Attainment should be rated at **75% or 3.75** on a 1—5 scale, or better.

Observation: Summer 2017: **5.00** Fall 2017: **4.94** Spring 2018: **5.00**

Summer 2018: **5.00** Fall 2018: **5.00** Spring 2019: **5.00**

Weighted over the entire assessment period (106 projects): 4.99

Evaluation: Graduating students consider this Student Outcome highly relevant, and more than 91% believe that they have attained it. Indicators 2 through 18 comfortably (rating of **Very High**) meet the acceptable threshold for the Value and the Coverage of Course Outcomes for all relevant courses (sole exception – Coverage in CAP 4630). For two of the six course-embedded assessments for relevant courses (Indicators 19 through 25) the students attained the desired level of proficiency (78.57 and 87.18% of students pass the criterion). We are frankly surprised by this result. Finally, our Senior Projects Assessment (Indicator 20) shows that the students have achieved the highest level of proficiency for this outcome. Although the Course-Embedded assessments do not meet our strict criteria, the performance of students in the Capstone Project is exceptionally good for this criteria. Attainment of Student Outcome (b) is rated as **very good**.

c) **An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.**

Indicators

1. Graduating Student Ratings	Relevance 96.31%	Attainment 87.3%	Sample: 109
2. Course Outcomes CAP 4104	Value 98.0%	Coverage: 96.0%	Sample: 58
3. Course Outcomes CAP 4630	Value 75.8%	Coverage: 67.6%	Sample: 24
4. Course Outcomes CAP 4641	Value 98.4%	Coverage: 95.6%	Sample: 36
5. Course Outcomes CAP 4770	Value 98.0%	Coverage: 91.8%	Sample: 36
6. Course Outcomes CDA 4625	Value 95.8%	Coverage: 78.4%	Sample: 24

7. Course Outcomes CEN 4010	Value 95.6%	Coverage: 89.2%	Sample: 58
8. Course Outcomes CEN 4021	Value 97.4%	Coverage: 98.0%	Sample: 17
9. Course Outcomes CEN 4072	Value 91.6%	Coverage: 84.2%	Sample: 42
10. Course Outcomes CNT 4713	Value 94.6%	Coverage: 88.0%	Sample: 102
11. Course Outcomes COP 2210	Value: 92.2%	Coverage: 87.8%	Sample: 173
12. Course Outcomes COP 3337	Value: 88.4%	Coverage: 81.6%	Sample: 123
13. Course Outcomes COP 3530	Value: 91.0%	Coverage: 88.4%	Sample: 97
14. Course Outcomes COP 4226	Value: 89.8%	Coverage: 90.4%	Sample: 17
15. Course Outcomes COP 4338	Value: 91.6%	Coverage: 81.8%	Sample: 96
16. Course Outcomes COP 4610	Value: 93.8%	Coverage: 88.8%	Sample: 90

17. Course-Embedded Assessment – CEN 4010

Spring 2018 Event: 24 students were evaluated for their Project Documentation.

Criterion: 75% of students should score at least 75% or higher.

Observation: 54.17% of the students answered at least 8 questions correctly.

18. Course-Embedded Assessment – COP 3337 (Inheritance)

Fall 2017 Event: 43 students completed an 8-question multiple choice assessment quiz.

Criterion: 75% of students should score at least 6 points.

Observation: 90.70% (37 out of 43) of the students answered at least 6 questions correctly.

19. Course-Embedded Assessment – COP 3337 (Exceptions)

Fall 2017 Event: 43 students completed an 8-question multiple choice assessment quiz.

Criterion: 75% of students should score at least 6 points.

Observation: 88.37% of the students (38 out of 43) answered at least 6 questions correctly.

20. Course-Embedded Assessment – COP 3530 (Abstraction)

Fall 2017 Event: 36 students completed an 8-question multiple choice assessment quiz.

Criterion: 75% of students should score at least 6 points.

Observation: 91.67% of the students (33 out of 36) answered at least 6 questions correctly.

21. Course-Embedded Assessment – COP 3530 (API Usage)

Fall 2017 Event: 36 students completed a 12-question multiple choice assessment quiz.
Criterion: 75% of students should score at least 9 points.
Observation: **86.11%** of the students (31 out of 36) answered at least 9 questions correctly.

22. Course-Embedded Assessment – COP 3530 (Linked Structures)

Fall 2017 Event: 36 students completed an 8-question multiple choice assessment quiz.
Criterion: 75% of students should score at least 6 points.
Observation: **66.67%** of the students (24 out of 36) answered at least 6 questions correctly.

23. Course-Embedded Assessment – COP 3530 (Recursion)

Fall 2017 Event: 36 students completed an 8-question multiple choice assessment quiz.
Criterion: 75% of students should score at least 6 points.
Observation: **86.11%** of the students (31 out of 36) answered at least 6 questions correctly.

24. Course-Embedded Assessment – COP 4338 (C Language)

Fall 2017 Event: 20 students completed a 10-question multiple choice assessment quiz.
Criterion: 75% of students should score at least 7.5 points.
Observation: **65.0%** of the students (13 out of 20) answered at least 8 questions correctly.

25. Senior Project Assessment

Event: Artifacts of all completed Senior Projects are assessed, by application of the *Senior Project Assessment of Student Outcomes of the BS in Computer Science* rubric, for attainment of outcome c). This event was replicated in all semesters from summer 2017 to spring 2019.

Criterion: Attainment should be rated at **75% or 3.75** on a 1—5 scale, or better.

Observation: Summer 2017: **4.00** Fall 2017: **3.97** Spring 2018: **4.00**

Summer 2018: **4.00** Fall 2018: **4.00** Spring 2019: **5.00**

Weighted over the entire assessment period (106 projects): 4.27

Evaluation: Except for the Coverage of outcomes in CAP 4630 (67.6%), and three course-embedded assessments, all indicators suggest that attainment of Student Outcome c) is very good.

d) Demonstrate the ability to work cooperatively in teams.

Indicators

1. Graduating Student Ratings	Relevance 95.38%	Attainment 90.80%	Sample: 109
2. Course Outcomes CEN 4010	Value: 95.6%	Coverage: 89.2%	Sample: 58
3. Course Outcomes CEN 4021	Value: 97.4%	Coverage: 98.0%	Sample: 17

4. Course Outcomes CIS 4911 Value: 93.8% Coverage: 85.2% Sample: 90
5. Senior Project Assessment

Event: Artifacts of all completed Senior Projects are assessed, by application of the *Senior Project Assessment of Student Outcomes of the BS in Computer Science* rubric, for attainment of outcome d). This event was replicated in all semesters from summer 2017 to spring 2019.

Criterion: Attainment should be rated at **75% or 3.75** on a 1—5 scale, or better.

Observation: Summer 2017: **4.78** Fall 2017: **4.75** Spring 2018: **4.65**
 Summer 2018: **5.00** Fall 2018: **4.72** Spring 2019: **4.82**
Weighted over the entire assessment period (106 projects): 4.77

Evaluation: All indicators suggest that attainment of Student Outcome d) is **excellent**.

- e) *An understanding of professional, ethical, legal, security and social issues and responsibilities*

Indicators

1. Graduating Student Ratings Relevance 87.32% Attainment 86.99% Sample: 109
2. Course Outcomes CGS 3095 Value: 91.6% Coverage: 90.2% Sample: 119
3. Course-Embedded Assessment CGS 3095 (Social and Ethical Concerns in Computing) – face-to-face section

Fall 2017 Event: Individual projects for 65 students were graded on a 4-point scale.

Criterion: 75% of students should score at least 3 points.

Observation: 89.23% of the students (58 out of 65) received at least 3 points.

4. Course-Embedded Assessment CGS 3095 (Social and Ethical Concerns in Computing) – Online section (RVC)

Summer 2017 Event: Individual projects for 92 students were graded on a 4-point scale.

Criterion: 75% of students should score at least 3 points.

Observation: 69.57% of the students (64 out of 92) received at least 3 points.

5. Senior Project Assessment

Event: Artifacts of all completed Senior Projects are assessed, by application of the *Senior Project Assessment of Student Outcomes of the BS in Computer Science* rubric, for attainment of outcome e). This event was replicated in all semesters from summer 2017 to spring 2019.

Criterion: Attainment should be rated at **75% or 3.75** on a 1—5 scale, or better.

Observation: Summer 2017: **3.00** Fall 2017: **3.00** Spring 2018: **3.00**
 Summer 2018: **3.00** Fall 2018: **3.00** Spring 2019: **3.00**
Weighted over the entire assessment period (106 projects): 3.00

Evaluation: Graduating students rate this outcome as extremely relevant and feel that they have attained it (Indicator 1). Current students find this outcome to be Highly Valuable and believe that it is Very-well covered in the classroom (Indicator 2). Evaluation of student projects in CGS 3095 show that students demonstrate **excellent** understanding of social and ethical issues in computing when the course is taught in face-to-face mode (Indicator 3), but demonstrate **good** rating in the online section of the course (Indicator 4). Senior project assessment (Indicator 4) shows that there is not much in student projects that evaluates these topics. On balance, attainment of Student Outcome e) is rated as **good**.

f) *An ability to communicate effectively with a range of audiences.*

Indicators

1. Graduating Student Ratings Relevance 94.3% Attainment 90.28% Sample: 109
2. Course Outcomes CGS 3095 Value: 91.6% Coverage: 90.2% Sample: 119
3. Course Outcomes CEN 4010 Value 95.6% Coverage: 89.2% Sample: 58
4. Course-Embedded Assessment CGS 3095 (Effective Communication Skills) – face-to-face section

Fall 2017 Event: Presentation of projects for 59 students were graded on a 4-point scale.

Criterion: 75% of students should score at least 3 points.

Observation: 100% of the students received at least 3 points.

5. Course-Embedded Assessment CGS 3095 (Effective Communication Skills) – Online section (RVC)

Summer 2017 Event: Presentation of projects for 92 students were graded on a 4-point scale.

Criterion: 75% of students should score at least 3 points.

Observation: 77.17% of the students (71 out of 92) received at least 3 points.

6. Senior Project Assessment

Event: Artifacts of all completed Senior Projects are assessed, by application of the *Senior Project Assessment of Student Outcomes of the BS in Computer Science* rubric, for attainment of outcome f). This event was replicated in all semesters from summer 2017 to spring 2019.

Criterion: Attainment should be rated at **75% or 3.75** on a 1—5 scale, or better.

Observation: Summer 2017: **5.00** Fall 2017: **5.00** Spring 2018: **5.00**
 Summer 2018: **5.00** Fall 2018: **5.00** Spring 2019: **5.00**

Weighted over the entire assessment period (106 projects): 5.00

Evaluation: All indicators suggest that attainment of Student Outcome f) is excellent.

- g) **An ability to analyze the local and global impact of computing on individuals, organizations, and society.**

Indicators

1. Graduating Student Ratings Relevance 86.91% Attainment 86.15% Sample: 84
2. Course Outcomes CGS 3095 Value: 91.6% Coverage: 90.2% Sample: 119
3. Course-Embedded Assessment CGS 3095 (Social and Ethical Concerns in Computing) – face-to-face section

Fall 2017 Event: Individual projects for 64 students were graded on a 4-point scale.

Criterion: 75% of students should score at least 3 points.

Observation: 100.0% of the students received at least 3 points.

4. Course-Embedded Assessment CGS 3095 (Effective Communication Skills) – Online section (RVC)

Summer 2017 Event: Presentation of projects for 92 students were graded on a 4-point scale.

Criterion: 75% of students should score at least 3 points.

Observation: 55.43% of the students (51 out of 92) received at least 3 points.

5. Senior Project Assessment

This outcome is not rated by the Senior Project course.

Evaluation: Graduating students rate this outcome as extremely relevant and feel that they have attained it (Indicator 1). Current students of CGS 3095 find this outcome to be Highly Valuable and believe that it is Very-well covered in the classroom (Indicator 2). Evaluation of student projects in CGS 3095, face-to-face section, show that students demonstrate **excellent** understanding of social and ethical issues in computing (Indicator 3) whereas it shows only satisfactory understanding for students in the Online section of the course (Indicator 4). Attainment of Student Outcome g) is rated as very good.

- h) **Recognition of the need for and an ability to engage in continuing professional development.**

Indicators

1. Graduating Student Ratings Relevance 94.98% Attainment 88.61% Sample: 84

2. Senior Project Assessment

Event: Artifacts of all completed Senior Projects are assessed, by application of the *Senior Project Assessment of Student Outcomes of the BS in Computer Science* rubric, for attainment of outcome h). This event was replicated in all semesters from summer 2017 to spring 2019.

Criterion: Attainment should be rated at **75% or 3.75** on a 1—5 scale, or better.

Observation: Summer 2017: **5.00** Fall 2017: **4.97** Spring 2018: **5.00**
Summer 2018: **5.00** Fall 2018: **5.00** Spring 2019: **5.00**

Weighted over the entire assessment period (106 projects): 5.00

Evaluation: All indicators suggest that attainment of Student Outcome h) is excellent.

i) **An ability to use current techniques, skills, and tools necessary for computing practice.**

Indicators

1. Graduating Student Ratings Relevance 95.55% Attainment 83.72% Sample: 108

2. Course-Embedded Assessment – CEN 4010

Spring 2018 Event: 24 students were evaluated for their Project Documentation.

Criterion: 75% of students should score at least 75% or higher.

Observation: **54.17%** (13 out of 24) of the students answered at least 8 questions correctly.

3. Senior Project Assessment

Event: Artifacts of all completed Senior Projects are assessed, by application of the *Senior Project Assessment of Student Outcomes of the BS in Computer Science* rubric, for attainment of outcome i). This event was replicated in all semesters from summer 2017 to spring 2019.

Criterion: Attainment should be rated at **75% or 3.75** on a 1—5 scale, or better.

Observation: Summer 2017: **5.00** Fall 2017: **5.00** Spring 2018: **5.00**
Summer 2018: **5.00** Fall 2018: **5.00** Spring 2019: **4.69**

Weighted over the entire assessment period (106 projects): 4.92

Evaluation: All indicators suggest that attainment of Student Outcome i) is excellent understanding that most of this is measured in the Senior Project.

j) **An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.**

Indicators

1. Graduating Student Ratings Relevance 92.65% Attainment 89.50% Sample: 82
2. Senior Project Assessment

Event: Artifacts of all completed Senior Projects are assessed, by application of the *Senior Project Assessment of Student Outcomes of the BS in Computer Science* rubric, for attainment of outcome j). This event was replicated in all semesters from summer 2017 to spring 2019.

Criterion: Attainment should be rated at **75% or 3.75** on a 1—5 scale, or better.

Observation: Summer 2017: **3.06** Fall 2017: **3.00** Spring 2018: **3.00**
 Summer 2018: **3.00** Fall 2018: **3.00** Spring 2019: **3.16**

Weighted over the entire assessment period (106 projects): 3.05

Evaluation: Existing students rate this outcome as extremely relevant and feel that they have attained it (Indicator 1). Students do not demonstrate attainment of this outcome in their Senior project, but this is indicative more of the non-application of theoretical concepts in their projects rather than their attainment. Overall, the attainment of Student Outcome j) is rated as **good**.

- k) An ability to apply design and development principles in the construction of software systems of varying complexity.**

Indicators

1. Graduating Student Ratings Relevance 95.68% Attainment 89.16% Sample: 107
2. Course Outcomes CEN 4010 Value: 95.6% Coverage: 89.2% Sample: 58
3. Course-Embedded Assessment – CEN 4010

Spring 2018 Event: 24 students were evaluated for their Project Documentation.

Criterion: 75% of students should score at least 75% or higher.

Observation: **54.17%** of the students answered at least 8 questions correctly.

4. Senior Project Assessment

Event: Artifacts of all completed Senior Projects are assessed, by application of the *Senior Project Assessment of Student Outcomes of the BS in Computer Science* rubric, for attainment of outcome k). This event was replicated in all semesters from summer 2017 to spring 2019.

Criterion: Attainment should be rated at **75% or 3.75** on a 1—5 scale, or better.

Observation: Summer 2017: **5.00** Fall 2017: **5.00** Spring 2018: **5.00**
 Summer 2018: **5.00** Fall 2018: **5.00** Spring 2019: **5.00**

Weighted over the entire assessment period (106 projects): 5.00

Evaluation: Indicators show that Attainment of Student Outcome k) is rated as **very good**.

C. Program Educational Objectives

The principal means of assessing attainment of the Program Educational Objectives of the BS in Computer Science program are the Alumni and Employer Surveys of Program Objectives.

The alumni responses are summarized in Table 4 (Section III.D) showing the averages of the 122 responses in the period from May 2019 to November 2019. The alumni responses provide ratings of the specific facets of each objective, and overall ratings of some objectives. The Alumni Survey raw data are included in Appendix E-1.

The employer survey (Appendix E-2) responses are summarized in Table 5 (Section III.E) showing the averages of the 5 responses in the period from May 2019 to November 2019. These responses provide ratings of specific facets of each objective and the overall rating of their combination. The relevant data is included in Appendix E-3.

Attainment of Student Outcomes enables attainment of the Program Educational Objectives, and so some Student Outcome data are again noted in this section where relevant. Additionally, the other constituent groups within the SCIS umbrella; WICS, ACM, STARS, UPE, Programming Team, and Industrial Advisory Board (IAB) may provide indicators of the attainment of the program objectives. The activity reports of the student organizations are included in Appendix H, and the minutes of the IAB meetings during the assessment period are included in Appendix I. Since the beginning of 2015, we hold two Board meetings per year.

1. Be successful in applying for entry level professional positions in computing-related fields, or for admission to graduate programs.

Indicators

- Alumni Survey of Program Educational Objectives:
Please rate how your educational experience at FIU contributed to your preparation for a career in computer science
May 2019 to October 2019: **77.78%** Previous cycle: **73.52%**
Please rate how your educational experience at FIU contributed to your preparation for graduate study
May 2019 to October 2019: **72.92%** Previous cycle: **73.66%**
- Employer Survey of Program Educational Objectives:
Please rate the following skill of our graduates: Will you consider employing our graduates in the future
May 2019 to November 2019: **100%** Previous Cycle: **91.00%**

This is at best a very indirect metric to gauge the overall attainment of this Program Objective from the employers' viewpoint.

- Student Chapter activities (Appendix H): Students are given plenty of opportunity to participate in a variety of activities that improve their association with the computing communities and learn the field outside the classroom. Some examples are given below.

ACM Chapter: Organized Workshops and Hack-a-thons (HackRiddle, MangoHacks), Coding Activities; helped students secure internships, improve their resume writing skills, taught them how to use Git version control in order to learn how to interview with companies; launched undergraduate research program.

WICS: Organized Workshops, activities like Hacking with Alexa and ShellHacks Breaking the Glass Ceiling Challenge, providing opportunity to students to participate in Grace Hopper Celebration Event with financial support from SCIS.

Upsilon Pi Epsilon: Organized Technical Workshops, Information Sessions on many topics, conducted the Google igniteCS Program to give students the opportunity to promote computer science education and reach out to the community, participated in collaborative projects with other student organizations.

STARS: Provided High Quality peer-to-peer tutoring services, provided Voluntary tutoring services for non-majors.

Programming Team activities: Programming Team members received weekly tutorials, training sessions, weekly mock competitions, travel to attend coaching camps and retreats, and master classes by visiting expert coaches; Team members were recommended and received internship opportunities at Ultimate Software, Google, Apple, Uber, and more, where many have become full-time employees; Members were provided scholarships; the team head coach took a group of FIU team members to a highly selective competition problem solving workshop in Spain taught by the coaches of the world's best programming teams from Russia.

Evaluation: Employers truly like the training provided to our students, and overwhelmingly indicate that they will continue to hire them. Our alumni observe that they are well equipped for their professional careers after graduation. And our student chapters are doing exceedingly well in holding workshops on a variety of topics of interest to their membership and providing them an opportunity to learn about new topics and participating in newer academic activities. Attainment of Program Educational Objective 1 is deemed **acceptable** with a rating of **very good**.

2.1 Be prepared for career accomplishment, responsibility and advancement in computing-related professions by virtue of having received in the BS program, a high-quality technical education in computing.

Indicators

- Alumni Survey of Program Educational Objectives:
Please rate the quality of your preparation upon graduation in Computer Programming
 May 2019 to October 2019: **84.03%** Previous cycle: **79.09%**
Please rate the quality of your preparation upon graduation in Systems Development

May 2019 to October 2019: **69.44%** Previous cycle: **72.17%**
Please rate the quality of your preparation upon graduation in Data Structures & Algorithms
 May 2019 to October 2019:: **81.25%** Previous cycle: **80.25%**
Please rate the quality of your preparation upon graduation in Computer Architecture & Organization
 May 2019 to October 2019: **71.53%** Previous cycle: **74.06%**
Calculated Overall rating of Technical Preparation upon Graduation
 May 2019 to October 2019: **76.56%** Previous cycle: **76.38%**

- Employer Survey of Program Educational Objectives:
Please rate the following skill of our graduates: Mastery of the fundamental computer science concepts and ability to solve computing problems using them
 May 2019 to October 2019: **85.00%** Previous Cycle: **77.50%**
- Enabling Student Outcomes – Graduating Student Survey:
 - a) A - Ability to apply knowledge of computing and mathematics: **90.54%**
 - b) B - Ability to analyze problem - identify and define its computing requirements: **91.23%**
 - c) C - Ability to design, implement, and evaluate a computer-based system: **87.30%**
 - d) I - Ability to use current techniques, skills, and tools necessary for computing practice: **83.72%**
 - e) J - Ability to apply mathematical foundations and algorithmic principles in design of computer systems: **89.50%**
 - f) K - Ability to apply design and development principles to construct complex software systems: **89.16%**

Evaluation: This Program Educational Objective is paramount. The percentage ratings shown above for the current Alumni survey cycle are almost the same as those reported in the 2017 assessment report:

<u>Alumni Survey Period</u>	<u>5/19 to 10/19</u>	<u>5/17 to 10/17</u>
# Responses	122	169
Computer Programming	84.03	79.09
Systems Development	69.44	72.17
Data Structures & Algorithms	81.25	80.25
Architecture & Organization	71.53	74.06

The ratings for preparation in the Systems Development and Computer Organization & Architecture areas have been consistently slightly below acceptable while the ratings for Data Structures & Algorithms and Computer programming have consistently been high. Note that ratings are better than those received in the last assessment cycle. *It is interesting to note that when they graduate, the students feel that they have attained proficiency in essentially all CS areas at a very high rating, but it diminishes considerably when they have worked in the industry for a while.*

Attainment of Program Educational Objective 2.1 is deemed **acceptable** with a rating of **very good**.

2.2 Be prepared for career accomplishment, responsibility and advancement in computing-related professions by virtue of having received in the BS program, communication and team-work skills.

Indicators

- Alumni Survey of Program Educational Objectives:
Please rate how your educational experience at FIU contributed to the development of your communication skills
May 2019 to October 2019: **77.78%** Previous cycle: **75.89%**

- Employer Survey of Program Educational Objectives:
Please rate the following skill of our graduates: Ability to communicate verbally
May 2019 to October 2019: **95.00%** Previous Cycle: **72.75%**
Please rate the following skill of our graduates: Ability to communicate in written form
May 2019 to October 2019: **95.00%** Previous Cycle: **59.00%**
Please rate the following skill of our graduates: Ability to work cooperatively in teams
May 2019 to October 2019: **95.00%** Previous Cycle: **82.50%**

- Enabling Student Outcomes – Graduating Student Rating:
 - a) D - Ability to function effectively on teams to accomplish a common goal: **90.80**
 - b) F - Ability to communicate effectively with a range of audiences: **90.28%**

Evaluation: As for the previous Program Outcome, it is interesting that the perspective on this outcome/objective should differ in the interim from graduation to employment. While the enabling outcomes are rated as high by seniors and evaluated to be excellent by employers (low response rate), the alumni assign only acceptable rating. This circumstance underscores the need to have continuing communication and dialog with our alumni.

Attainment of Program Educational Objective 2.2 is deemed **acceptable** with a rating of **very good**.

2.3 Be prepared for career accomplishment, responsibility and advancement in computing-related professions by virtue of having received in the BS program, awareness of the ethical and social responsibilities of their profession.

Indicators

- Alumni Survey of Program Educational Objectives:
Please rate how your educational experience at FIU contributed to the development of your awareness of social and ethical responsibility
May 2019 to October 2019: **76.39%** Previous cycle: **73.82%**

- Employer Survey of Program Educational Objectives:
Please rate the following skill of our graduates: Understanding of Social and Ethical Concerns

May 2019 to October 2019: **87.50%** Previous Cycle: **72.50%**

- Enabling Student Outcomes – Graduating Student Rating:
 - a) E - Understanding of professional, ethical, legal, security, and social issues: **86.99%**
 - b) G - Ability to analyze local and global impact of computing on society: **86.15%**

Evaluation: As for the previous Program Outcome, it is interesting that the perspective on this outcome/objective should differ in the interim from graduation to employment. While the enabling outcomes are rated as very high by seniors and employers, the alumni assign only a high rating. It is reasonable to ascribe the adjustment to the real-world experiences of our graduates, but this is a conjecture.

Attainment of Program Educational Objective 2.3 is deemed **acceptable** with a rating of **good**.

2.4 Be prepared for career accomplishment, responsibility and advancement in computing-related professions by virtue of having received in the BS program, an ability to engage in continued professional development activities.

Indicators

- Alumni Survey of Program Educational Objectives:

Please rate how your educational experience at FIU contributed to your capacity for personal growth

May 2019 to October 2019: **84.72%** Previous cycle: **81.36%**

Please rate how your educational experience at FIU contributed to your capacity for lifelong learning

May 2019 to October 2019: **84.72%** Previous cycle: **82.84%**

- Employer Survey of Program Educational Objectives:

Please rate the following skill of our graduates: Ability to learn new and Emerging Concepts and Technologies

May 2019 to October 2019: **90.00%** Previous Cycle: **75.00%**

- ACM Chapter activities (Appendix H)
 - ❖ Organization of student-led, faculty-led, and industry-led Workshops
 - ❖ Coding activities (HackRiddle, Mango Hacks, etc.)
 - ❖ helping students secure internships, improve their resume writing skills, teaching them how to use Git version control in order to learn how to interview with companies
 - ❖ Organizing Social Events
 - ❖ In addition to these activities in this assessment period, ACM Student Chapter launched Undergraduate Student Research Program.
- UPE Activities (Appendix H)
 - ❖ Organization of technical Workshops

- ❖ Organizing Information Sessions (Software, Hardware, and Cyber Development Programs)
 - ❖ Conducted other Activities (Google Ignite CS, Gaming Tournament, FIU Relay for Life, MangoHacks, ShellHacks, Engineering Expo, and others)
 - ❖ Google igniteCS Program continues to give students the opportunity to promote computer science education and reach out to the community. Through the program, students currently visit 15 elementary and middle schools in Miami-Dade County every week, teaching over 500 students, computer science.
 - ❖ Collaborative projects with other student organizations in SCIS and FIU (MentorFIU, SparkDev, Discord HypeSquad, etc.)
 - ❖ Organizing many social events
 - ❖ Organizing a Town Hall Meeting between students, School administrators, and faculty
- WICS Activities (Appendix H)
 - ❖ Participating in Grace Hopper Celebration every year
 - ❖ Organizing Workshops (Virtual Reality, Google G-Suite, Algorithm Practice, Soldering, Tech Summer Camp, and so on)
 - ❖ Holding various academic events (Gaming Tournament with UPE, CodeFest Big Sister Mentorship, MangoHacks Ladies Storm Hackathons, Hacking with Amazon Alexa, Girls Who Code Panel, etc.)
 - ❖ Organizing social events
- STARS Activities (Appendix H)
 - ❖ Providing high quality Peer Tutoring Services for many courses with primary focus on Java programming, Data Structures, Databases, and Networking
 - ❖ Volunteer Peer Tutoring for non-major students in CGS 2060, CGS 2100, and CGS 2518.
 - ❖ Scheduling Midterm and Final Exam Review Sessions
 - ❖ In Summer 2019 STARS offered fully online tutoring services via the use of WhatsApp chat groups. This in support of our online offerings served a very useful purpose deemed extremely helpful by online students
- Programming Team Activities (Appendix H)
 - ❖ The teams have received scholarships, weekly tutorials, training sessions, weekly mock competitions, travel to attend coaching camps and retreats, and master classes by visiting expert coaches
 - ❖ Most programming team member have served an internship at Ultimate Software, Google, Apple, Uber, and more. Many have since become full time employees at their interning companies
 - ❖ Other team members have enrolled in graduate studies
 - ❖ The team head coach took a group of FIU team members to a highly selective competition problem solving workshop in Spain taught by the coaches of the world's best programming teams from Russia
 - ❖ Scholarships for Team Members were provided in 2017-2018 (\$17,480) as well as 2018-2019 (\$14,000).

- ❖ In the summer of 2019, a 5-day training camp for competition problem solving was held on the campus of FIU in partnership with the programming team from UNAL, Bogota, Colombia
- ❖ Team members have participated in ACM Regional Programming Competition for many years. In particular, participating in Division 2 in 2019, the team ranked 2nd.
- ❖ In Spring of each year, the Academy hosts the Annual **FIU High School Programming Competition**, attended by about 40 teams from Florida high schools, the largest competition of its kind in South Florida.

Evaluation: It is not clear that attainment of this objective is directly enabled by specific courses in the Computer Science major. Rather, it is the collective breadth represented by the entire BS in Computer Science program that may have an enabling effect. In addition, the breadth component common to all FIU majors, the Core Curriculum and non-major elective courses, is a principal contributor to any graduated student's realization of personal growth and capacity for life-long learning.

Alumni clearly feel that their education at FIU contributed greatly to their personal growth and lifelong learning experiences.

Involvement with the School's student organizations is another excellent enabler of this objective, but these experiences are voluntary and are not exploited by a majority of our graduates, particularly night students.

Attainment of Program Educational Objective 2.4 is deemed **acceptable** with a rating of **very good**.

V. RECOMMENDATIONS

A. Recommendations of the Subject Area Coordinators

Subject Area: Applications (SAC: Mark Finlayson)

CAP 4104: No changes are recommended.

CAP 4453: This course was not offered in this evaluation period – No changes are recommended.

CAP 4630: Students complained that the course covered too much material, and the SAC agrees with them. Accordingly, the following course topic changes are recommended: (1) Remove the unit on game theory and multi agent systems, (2) Remove mixed integer programming, linear programming, and MDPs from unit two, and (3) Spread the remaining material across the allotted time.

CAP 4641: No changes are recommended.

CDA 4625: Include more hands-on and visual material as suggested by students.

Subject Area: Professional Development (SAC: Richard Whitaker)

CGS 1920: No changes are recommended.

CGS 3095: No changes are recommended. However, some students requested that the course material include more focus on the impacts of social media and destructiveness of tech startups.

ENC 3249: No changes are recommended. However, technical writing should be emphasized more in this course as instructors found the students “deficient to adequate” in writing skills.

Subject Area: Computer Organization (SAC: Nagarajan Prabakar)

CDA 3103: It was observed that the use of interactive textbooks (Zybooks) improves student learning, and it should be continued in the new course CDA 3102.

CDA 4101: An introductory lecture with online resources for Verilog at the beginning of the term is essential. For each group project, include peer evaluations among group members to address an important concern expressed by students about sharing the work in group projects and the credit for each team member.

CNT 4713: No change is needed on the course outcomes or syllabus.

COP 4610: No change is needed on the course outcomes or syllabus.

Subject Area: Computer Systems (SAC: Jason Liu)

CAP 4612: The course was not offered during this evaluation cycle, and hence, no changes are recommended.

CAP 4710: No changes are recommended.

CAP 4770: No changes are recommended.

CEN 4083: No changes are recommended.

COP 4604: No changes are recommended.

COP 4710: No changes are recommended.

COP 4722: One instructor consistently indicated that the objective “Object-Oriented Database” and “Spatial and Multimedia Databases” as inappropriate. It is recommended that two outcomes of this course need to be discussed and possibly readjusted.

Subject Area: Foundations (SAC: Xudong He)

CAP 4506: This course was only offered once in the past two years. No changes are recommended.

COP 4534: Instructors found students’ preparedness for the class to be between deficient and adequate. An instructor comment suggested that students should have some basic knowledge of combinatorics, statistics, and probability before taking this course. This lack of preparation is observed in many courses in this subject area, and we need to come up with some scheme of topic coverage in this and pre-requisite courses to alleviate this problem.

COP 4555: Instructors found that the student preparation for this class ranges from deficient to adequate. A few professor appraisal comments stated that students need better mathematics preparation to understand the essential concepts of functions, sets, and relations; better rigorous thinking and logical reasoning capabilities; and that the course be taught in a laboratory to practice programming in F#. Lack of preparation is observed in many courses in this subject area, and we need to come up with some scheme of topic coverage in this and pre-requisite courses to alleviate this problem.

COT 3100: Students’ preparation for this course ranges from non-existent, deficient, adequate to good. A few professor appraisal comments included (1) student must develop stronger work ethics prior to enrolling in this course, (2) the number of course objectives is too high, (3) the outcomes related to programming should be compressed into a single outcome and make it be “familiarity” rather than implementation, (4) students have a very low level of math and logical reasoning and therefore it is very difficult for them to formalize problems and proofs, and (5) there is no time to properly cover some of the objectives related to program implementation. We need to address the lack of preparation for this course, too.

COT 3541: Students suggested that more videos be used to explain the course material, to have more time on Prolog, to have a better textbook, and to have more examples. One comment of the online offering was to change discussion posts to classwork.

Students' preparation for this course was adequate. A few professor appraisal comments included this course has effectively challenged students to think and logic provides the unifying foundation for computer science. One suggestion was to explicitly cover propositional logic to help students have a consistent and systematic knowledge of various concepts in logic.

COT 4521: Instructor suggested that Data Structures and Linear Algebra should be required pre-requisites. Further, more demos could also help student understanding.

A general note: There are a few common problems in the Foundation Area courses listed above: (1) deficiency of students' preparation in math and logical thinking and (2) how to help students to better understand course materials and prepare for exams. The offering of COT-3100 discrete structures may alleviate problem (1) for some other courses, but itself encounters the same problem. To address problem (2), homework grading criteria need to be changed to discourage homework copying and encourage student efforts; and in-class practices and quizzes are used to improve students understanding of fundamental concepts and performance on exams. Several observations include low student evaluation responses and missing appraisal comments from several instructors consistently, which need to be addressed to improve learning.

MAD 2104 & MAD 3512: Neither student evaluations nor instructor appraisals are available for these courses. No changes are recommended.

Subject Area: Programming (SAC: Tim Downey)

COP 2210: Continue to evaluate the effectiveness of the math prerequisite, but do not change the prerequisite at this time. Continue to urge instructors to cover all the outcomes of the course. No change is needed on the course outcomes or syllabus.

COP 3337: The school has instituted a new design for COP2210, with fewer sections and a common exam. This should address the concern of students having diverse preparation for the course. The low coverage in some semesters is not a problem with the structure of the course, but with the presentation of the material. All instructors should be encouraged to cover all the material in a meaningful way. A review of the outcomes should be made to assess if removing some of the outcomes would maintain the content of the course and allow more time for other topics.

COP 3530: COP3337 instructors should ensure that all course outcomes are met. No change is needed in the course outcomes or syllabus.

COP 4226: Remove database connectivity from the outcomes because a database course is not a pre-requisite for this course.

COP 4338: The low coverage in some semesters mentioned by students is not a problem with the structure of the course, but with the presentation of the material. All instructors should be encouraged to cover all the material in a meaningful way. Instructors should be asked if there is enough time to cover the advanced material in the course while providing introductions to UNIX, pointers, and C data structures.

COP 4520: Neither the course outcomes nor the syllabus needs any changes.

Subject Area: Software Engineering (SAC: Masoud Sadjadi)

CEN4010:

- Observations:
 - The irrelevance of Net-Centric course as one of the pre-requisites for this course is rightfully questioned by the instructor and students of the course.
 - There is a request for adding agile software development approaches to this course to better serve the senior project.
 - Lack of enough teamwork experience is evident in some cases. Our professors would like our students to perform better in their groups.
 - Lack of enough exposure to software development tools such as version control (e.g., git).
 - Our students expect to learn more about the real-world problems and the state-of-the-art software engineering practices being used in industry.
 - They do not want to be bugged down with plenty of homework assignments and extra documentations that would be of no use to them in the future.
- Recommendations:
 - Prerequisite and Preparedness
 - ❖ Net-Centric should be removed from the list of prerequisites for this course.
 - ❖ Opportunities for teamwork experience in prior courses should be explored.
 - ❖ Opportunities to expose students to software development tools such as version control should be explored in prior courses.
 - Agile and Scrum software development approaches should be included in the syllabus of this course.
 - ❖ State-of-the-art practices of software development from industry should be adopted in this course.
 - ❖ An Agile/Scrum textbook should be included as a reference, if not the main textbook of the course.
 - ❖ Class lecture times should be spent more on practicing agile software engineering development than just giving lectures.
 - Learning by example and practice is the best way to transfer the knowledge and experience from the professor to the students.

CEN 4021: The lack of UML knowledge is an indication that some professors might have not put enough emphasis on learning and practicing UML diagrams in CEN 4010 for the sake of

adding some Agile/Scrum concepts. This should not be the case. Adding agile is a great improvement to CEN 4010, but it should not mean dropping the ball on the UML diagrams.

CEN 4072: The following recommendations are made.

- Bring the syllabus of this course up to speed with the state-of-the-art practices in industry, test-driven development is one of the popular agile software development practices in industry. Students should be exposed to this approach.
- Debugging should stay in the syllabus as testing without debugging would not help with improving the quality of the software solution.
- Give students some hands-on experience, a good portion of the lectures time should be spent more on practicing the testing/debugging methods using state-of-the-art tools. Alternatively, some online tutorials can be suggested to the students to do some self-learning.

CIS 4911: The following observations and recommendations are made by the SAC.

- Students should be better prepared for this class.
 - Add Agile/Scrum software development approaches to CEN 4010. Also, they should learn and experience how to be a good team member in a self-organizing Agile/Scrum development team.
 - Adding Agile to the syllabus of CEN 4010 should NOT mean that learning of UML diagrams should be dropped or taken lightly. Our students must know how to read/create the most popular UML diagrams. Unfortunately, this is not the case for many of our students.
 - Provide students with a compressed Agile/Scrum online training at the beginning of the semester so that those of them who are lacking some knowledge in this area can catch up before the work on their senior projects starts.
 - An eligibility test should be taken at the beginning of the semester so that students are well prepared to perform in a project. This would avoid issues with their teammates during the semester.
- The product owners should be better prepared for this class.
 - The product owners of approved projects must go through a short crash course on how to be a good product owner for our students.
 - They must commit to be available to answer our students' questions daily and be available to review/evaluate their work every other week and provide them with enough work for the following sprints ahead of time.
- Expectations from the students should be clearly communicated to them.
 - The instructor of the class must provide clear breakdown of the points and provide students with bi-weekly updates on their status.
 - More in-depth feedback should be provided to the students both by the product owner and the instructor of the class on an ongoing basis and when requested specifically by the students.

- All the requirements and guidance for the class should be easily accessible by the students. Even if some requirements and guidance may be required to be in different systems, there must be one starting point from which everything is accessible.
- Need for professional system staff support.
 - The project management tools adopted for this class in some cases had been hacked and the server went down.
 - There should be one or more system staff at SCIS assigned to this course to manage the support software tools for the students.

B. Recommendations of the Assessments Coordinator

1. Course Related:

AC-01: The Course Outcomes Surveys for MAD 3512 are not conducted. This is a continuing concern. If possible, some other assessment mechanism should be employed for MAD 3512 on a regular schedule. Further, only 10 out of 33 students (36.36%) performed at an expected level or slightly below. We need to examine the reasons for this inadequate performance of students, and then coordinate the content and delivery of this course better with the Department of Mathematics and Statistics.

AC-02: As expected in this assessment cycle, course outcomes for MAD 2104 are substituted by those for COT 3100, the course offered and controlled by SCIS. In the Course Embedded Direct Assessment for COT 3100, 22 out of 29 students (75.86%) demonstrated proficiency in Discrete Structures. This is significantly better than the evaluation for MAD 2104 (20 out of 36 = 55.56%). Students recommend to be assigned more in-class practice problems that should be considered by the instructors as a committee.

AC-03: All courses in the newly created Subject Area “Applications” except one require no changes. Artificial Intelligence (CAP 4630) was taught by an instructor who is no longer with us, and he did not do a good job at all. The course bites too much, and needs to be simplified as suggested by the SAC.

AC-04: In this assessment cycle, a Math pre-requisite was introduced for the first Programming Course, COP 2100. Students still seem to be deficient in their mathematical preparation for this course. It is suggested that a committee of instructors examine the effectiveness of the math pre-requisite.

AC-05: Students in the second Programming Course, COP 3337, were found deficient in their overall preparation for the course. The School has instituted a new design for the first Programming Course, COP 2210, with fewer sections and a common exam. This should begin addressing this concern. It is suggested that a committee of instructors examine the Course Outcomes to assess if removing some of the outcomes would maintain the content of the course and allow more time for other topics. The instructors should ensure that all Course Outcomes are met.

AC-06: For COP 4226, one of the Course Outcomes includes “database connectivity” when no database course is pre-requisite for this course. Remove “database connectivity” from the Course Outcomes.

AC-07: For the series of courses in the Subject Area “Foundations (COT 3100, COT 3541, COT 4521),” students’ preparation in mathematical and logical thinking was observed to be deficient. I suggest that we wait for the next assessment cycle to gauge the effect of COT 3100 towards this pre-requisite preparation of students for these courses. Also, it seems that students require increased assistance to better understand the course material and prepare for exams. It is suggested that more in-class practice problems be assigned to students along with quizzes to gauge their level of understanding.

AC-08: It appears that there is mixed feeling among instructors of COP 4722 about including “Object-Oriented Database” and “Spatial and Multimedia Databases” in the Course Outcomes. The SAC suggests that this be discussed and possible adjusted; I concur with that suggestion.

AC-09: To address student and instructor concerns for courses in Subject Area “Software Engineering,” the following suggestions are made for CEN 4010.

- Prerequisite and Preparedness
 - ❖ Net-Centric should be removed from the list of prerequisites for this course.
 - ❖ Opportunities for teamwork experience in prior courses should be explored.
 - ❖ Opportunities to expose students to software development tools such as version control should be explored in prior courses.
- Agile and Scrum software development approaches should be included in the syllabus of this course.
- State-of-the-art practices of software development from industry should be adopted in this course.
- An Agile/Scrum textbook should be included as a reference, if not the main textbook of the course.

AC-10: As suggested by some instructors of CEN 4072 “the details of debugging” is removed from the Course Outcomes. Some instructors believe that Debugging should stay in the syllabus as testing without debugging would not help with improving the quality of the software solution. I suggest that the instructors of this course come together and take a firm action on this issue. It is also suggested that students should be exposed to the test-driven development which is one of the most popular agile software development practices in the field.

AC-11: Most of the projects undertaken in CIS 4911 use Scrum. Accordingly, Scrum should be used in the pre-requisite course CEN 4010, or at the least, reviewed in CIS 4911. Additionally, students should be taught how to be a good team member for team projects.

AC-12: Beginning in Spring 2020, two courses, CDA 3103 and CDA 4101, will be replaced by CDA 3102. Since the use of interactive textbooks (Zybooks) was helpful in improving student learning in CDA 3103, it is suggested that its use be continued in CDA 3102 also.

2. Procedure Related:

AC-13: For a few courses, the Instructor Course Appraisals are not filed in. The Associate Director (or designee) should ascertain that these are filled by the instructors every term.

AC-14: We have now used the Employer Survey to measure attainment of Program Educational Objectives of our students for the third time. This is wonderful. However, the number of response (5 responses per question although 9 participated) was very low. It is recommended that meaningful steps be taken in the future to increase this response rate. This is a continuing concern.

AC-15: In this assessment cycle, student participation in the Course Evaluation System was quite poor. This is mainly due to discontinuing our practice of taking netbook computers to every class and making students fill in these surveys. If that is not doable now for difficulties in its implementation, then we must find other mechanism to improve this participation. May be, we should look into giving students some incentive to complete these surveys.

AC-16: Very soon in the future, we will need to supply data to ABET for their evaluation of the online degree, BA in Computer Science, along with continuing application for BS in CS degree. Accordingly, beginning now, we should consider collecting Student and Instructor Evaluation of Course Outcomes separately for our online offerings.

3. General:

AC-17: It is challenging to perform meaningful assessment of Student Outcome a) *Ability to apply knowledge of Computing and Mathematics* using the rubric of the Senior Project class because there are essentially no projects attempted by students that address the relevant topics. The point is made for discussion only; no recommendations are made.

AC-18: The quality, the variety, and the number of activities performed by our student clubs (teaching new subjects through workshops, providing opportunities to improve through technical activities, providing outreach to communities by helping students in middle and high schools, and so on) has increased by a substantial percentage as compared with their past activities. Programming Team is well supported by Ultimate Software for many years and by FaceBook starting in 2019. SCIS administrators should continue to support them in whatever way possible, including providing more space for their activities.

AC-19: For a few years now, the meeting of the Industrial Advisory Board is conducted at the end of the Fall and Spring semesters when selected students present their Capstone Projects. Members have been suitably impressed with their work in the past and continue to be impressed now. Many members act as mentors and/or judges for these projects. This has proven to be very beneficial for the students. We should continue to find more and better ways to engage the Board members in student activities.

AC-20: To improve the response rate of Employers in their survey, we need to engage the Industrial Board members to respond themselves as they do hire many of our graduates, and entice other employers through their connections. May be, a letter from the Chairperson of the Board, specifying the importance of participation in this survey, should be attached with the survey instrument.

VI. CONCLUSION

The BS in Computer Science program continues to deliver high quality preparation for entry into the computing work-force, or admission to graduate programs in computing. The delivery of its required coursework continues to receive very high ratings from students as expressed in the Surveys of Course Outcomes (4.34/5, 86.8%, Table 1). Evaluations of attainment of its Student Outcomes (88.6 %, Table 3) and Program Educational Objectives (3.12/4, 78.0%, Table 4) uniformly meet or exceed the minimum acceptability criteria.

In continuing to strive to ensure students' educational experiences are relevant to the reality of the work-force they enter, SCIS continues to offer capstone experience in the Senior Project course. This has been a phenomenal success as indicated by the students (Value of Outcome 4.69/5, 93.8%, Table 1) and showed in the measurements using the rubrics for that course. We continue to improve our offerings in many of our focus areas. The evaluation of Student Learning by various topics as part of the Course Embedded Direct Assessment Mechanism in many courses (COP 3337, COP 3530, COP 4338, and COP 4610) is providing us useful information to fine-tune our curriculum. MAD 3104 (Discrete Mathematics taught by Math Faculty) was substituted by COT 3100 (Discrete Structures taught by CS Faculty) in this assessment cycle, and the result has been excellent. Student indicated Value of Outcomes (4.32, 86.4%) and Coverage Adequacy (4.34, 86.8%) exceed our acceptable criteria.

Our course offerings have diversified (continued process) with the computing field's emphasis on new applications. Accordingly, in this Assessment cycle, we have added a new Subject Area to our list, "Applications." Other meaningful curriculum changes for the BS in CS Program will be installed beginning in Spring 2020 semester (moving some core courses to the list of electives, consolidating multiple courses into a single course to avoid duplication of topics covered and streamlining the courses better, and so on).

The student chapters have increased their activities in quality, quantity, as well as variety. The FIU-ACM student chapter was the winner of the 2018-2019 ACM Student Chapter Excellence Award in Chapter Activities for the second year in a row. In addition to its normal activities, it launched Undergraduate Student Research program this year. The Programming Team was ranked 2nd (Division 2) in 2018 ACM Regional Programming Competition. Three teams competed in 2019 at the same competition and were ranked 17th, 19th, and 25th. WICS, STARS, and UPE student chapters have become very active (Appendix H) and continue expanding their sphere of influence among participating members. It is noteworthy that our industry partners Ultimate Software, IBM, State Farm, and others offer many opportunities to students to learn about industry jobs, skills necessary to be successful once they enter the workforce, and the importance of problem solving through their engagement in various student activities like support of Programming Teams, mentorship and/or evaluation of capstone projects, and the like. WICS has been partnering with the other student groups for the benefit of its membership.

The SCIS Industrial Advisory Board is increasingly involved in all facets of our operation as indicated by the minutes of its meetings included in Appendix I. In particular, selected presentations of the Senior Projects are made to the Board by student teams, and members have praised the student work profusely. One Board member is very heavily involved in the evaluation

of every project. The overall involvement of the Board will be enhanced in the future; both the Board members and the SCIS administration desire it very much.

In the last few years, and particularly in this assessment period, we have substantially increased student participation in internships and employer participation in job fairs on campus. With a full-time staff member looking after these activities, the student success has been nothing short of phenomenal.

The biennial assessment is working out exceedingly well, and gives us more meaningful information from one report to the next. The participation of the entire faculty is serious and meaningful, and we observe good modifications in our curriculum based on the recommendations made in this report.

The ABET Review of the BS in CS Program was conducted in Fall 2016, and we passed it with flying colors. Looking at the results of these biennial assessments and using the recommendations of various constituencies to improve our curriculum and student learning, we sincerely believe that we are well-placed to succeed in the next ABET Review as well.