CENTRAL NEW MEXICO COMMUNITY COLLEGE
ASSESSMENT REPORT
Due to SAAC by October 11

PART 1: CONTACT & PROGRAM IDENTIFICATION

Report Year and Contact Information			
Fall 2016- Summer 2017	Ivonne Nelson, and Robert Garner	inelson1 and rgarner7	50270 and 52409
Academic Year	**Contact Person**	**Email**	**Phone Number**

Subject of this Assessment Report		
Program: Computer Information Systems, Computer Programming Concentration	**Gen Ed Area:**	**Non-Award, Non-Gen-Ed Discipline Area**:
☐ Certificate ☐ AA ☐ AS ☒ AAS	Applicable to: ☐ AA/AS ☐ AAS	

PART 2: THE YEAR IN RETROSPECT

Program/Area Highlights (Including, wherever applicable, course completion, job placement, and licensing examination information)
For the fiscal year 2016-17, of the 1,226 students with CIS declared major, 122 earned their degree. The course retention rate for all CIS was 86%. Successfully attained SLO's 1-6.

Changes Made in Support of Student Learning
Last year's action plan in support of student learning said: We will continue to emphasize debugging skills in the three languages, and we will bring in debugging skills earlier and more frequently in C++ I, C++ II, Java I and C#. We will add debugging practice in Android and ASP.net.   This has shown an maintenance of scores meeting targets in the debugging area. Although this cycle's success is "high enough", we will continue to emphasize this important Programming skill and aim for an improvement in level success next cycle.

PART 3: REPORT ON RECENT ASSESSMENT OF STUDENT LEARNING

Student Learning Outcome(s) Assessed *To add rows: right –click in cell below and select "Insert," "Insert Rows Above"*	Classes/Cohorts Assessed
1. Class construction: Write programs that contain a programmer-written class and demonstrate its use in the C++, Java and C# languages.	Outcomes 1-6 CIS Computer Programming students were assessed in their final semester via the CIS 2999 Capstone course.
2. Class inheritance, and polymorphism: Write a program that contains a programmer-written class structure including a parent class and at least two children classes. The program must demonstrate polymorphism.	
3. Graphical User Interface and Technical documentation: Write a program that contains a Graphical User Interface that includes event handling components. These components must include components such as menus, dialog boxes, sliders, buttons, and spinners. Tooltips must be on all components, where relevant. The program must contain a help section or additional documentation for the user.	
4. Database manipulation and Web Application: Write a program that demonstrates the ability to connect to and manipulate a SQL database.	
5. Web research: Use a search engine, such as "Google", to find information on classes or functions that are needed in a program. This web research includes finding the appropriate class/function, its documentation, and implementing the code in a program.	
6. Debugging: Demonstrate the use of a debugging tool in at least two Integrated Development Environments, with at least two languages.	
7. Linux: Students will demonstrate how to install, configure, create user accounts, issue correct commands and options, and perform standard network administration.	Outcome 7: All CIS concentrations which require Linux in their program, will report Linux assessment results. This assessment information reflects all CIS students who take the Linux course.

Measurement Tool(s) Used *To add rows: right –click in cell below and select "Insert," "Insert Rows Above"*	Internal	External	Direct	Indirect	Initial Achievement Target or Expectation
			Enter X's for type of tool		
Outcomes 1-5 were assessed via a portfolio of work created by the students. The students were given a detailed document describing the required computer programming topics. The document /portfolio topics were described and requested in order of the competencies. Students created a portfolio and each section addressed and demonstrated programming concepts.	X		X		Outcomes 1-6: The Computer Programming exit competencies are evaluated using a Rubrics with a scale of 4=excellent, 3=good, 2=fair and 1=poor. We believe a score of 3+ for 75% of our students represents success in accomplishing our goals.
Outcome 6 was assessed via a debugging exam given to students. The students were given three broken programs to debug , in C++, Java and C#.	X		X		Outcomes 1-6: The Computer Programming exit competencies are evaluated using a Rubrics with a scale of 4=excellent, 3=good, 2=fair and 1=poor. We believe a score of 3+ for 75% of our students represents success in accomplishing our goals.
Outcome 7 was assessed via the CIS 1680 Linux Essentials course. All students taking this course will be assigned a final project that encapsulates the exit competencies for this course	X		X		Outcome 7: Several CIS concentrations incorporate the Linux course in its area of studies. Our achievement target for all Linux students (for all concentrations requiring this course) is 80%+ on the assessment skills exam for 75% of our students.

Assessment Findings

	COMP 1	COMP 2	COMP 3	COMP 4	COMP 5	COMP 6
SCORE	CLASSES	INHERITANCE	GUI	DB MANIP	RESEARCH	DEBUG
4	16	18	5	14	15	0
3.5 – 3.9	7	4	11	6	3	11
3		2	4	3	6	8
2.5	2	1	2	1		6
2			3	1		
1.5						
1					1	
0					0	

	COMP 1	COMP 2	COMP 3	COMP 4	COMP 5	COMP 6
SCORE	CLASSES	INHERITANCE	GUI	DB MANIP	RESEARCH	DEBUG
3+	23	24	20	23	24	19
<3	2	1	5	2	1	6
Meet Target?	yes	yes	yes	yes	yes	yes

For Linux report, Overall, students who took CIS 1680 have;

TestOut Linux Pro Exam Pass Rate

   CNM... 77.42%
   National (All US) 72.74%

Analysis and Interpretation of Assessment Findings

Increased emphasis on Debugging skills showed improvements in student performance. We will continue this practice.

Action Plan in Support of Student Learning

We will continue to emphasize debugging skills in the three languages, and we will bring in debugging skills earlier and more frequently in C++ I, C++ II, Java I and C#. We will add debugging practice in Android and ASP.net.

Please indicate with an X all of the following that characterize the types of changes described in the above action plan:

[X] Pedagogical change   [ ] Course revision   [ ] Process revision   [ ] Curricular revision

[ ] Budgetary reallocation   [ ] Faculty training/development   [ ] Assessment criteria revision   [ ] Assessment methodology revision

Recommendations, Proposals, and/or Funding Requests
We will emphasize to all instructors the importance of debugging skills.

PART 4: ASSESSMENT CYCLE PLAN UPDATE (Copy and paste from original plan if unchanged)

Cycle Years	Description of Changes Made (if applicable)
	No Changes made.

Student Learning Outcomes	When Measured	Where Measured	How Measured
1. Class construction: Write programs that contain a programmer-written class and demonstrate its use in the C++, Java and C# languages.	Every semester of every year: 2016-2017	Program Portfolio Demonstration	Class construction: Write programs that contain a programmer-written class and demonstrate its use in the C++, Java and C# languages. Rubric used.
2. Class inheritance, and polymorphism: Write a program that contains a programmer-written class structure including a parent class and at least two children classes. The program must demonstrate polymorphism.	Every semester of every year: 2016-2017	Program Portfolio Demonstration	Class inheritance, and polymorphism: Write a program that contains a programmer-written class structure including a parent class and at least two children classes. The program must demonstrate polymorphism. Rubric used.
3. Graphical User Interface and Technical documentation: Write a program that contains a Graphical User Interface that includes event handling components. These components must include components such as menus, dialog boxes, sliders, buttons, and spinners. Tooltips must be on all components, where relevant. The program must contain a help section or additional documentation for the user.	Every semester of every year: 2016-2017	Program Portfolio Demonstration	Graphical User Interface and Technical documentation: Write a program that contains a Graphical User Interface that includes event handling components. These components must include components such as menus, dialog boxes, sliders, buttons, and spinners. Tooltips must be on all components, where relevant. The program must contain a help section or additional documentation for the user. Rubric used.
4. Database manipulation and Web Application: Write a program that demonstrates the ability to	Every semester of every	Program Portfolio	Database manipulation and Web Application: Write a program that

	connect to and manipulate a SQL database.	year: 2016-2017	Demonstration	demonstrates the ability to connect to and manipulate a SQL database. Rubric used.
5.	Web research: Use a search engine, such as "Google", to find information on classes or functions that are needed in a program. This web research includes finding the appropriate class/function, its documentation, and implementing the code in a program.	Every semester of every year: 2016-2017	Program Portfolio Demonstration	Web research: Use a search engine, such as "Google", to find information on classes or functions that are needed in a program. This web research includes finding the appropriate class/function, its documentation, and implementing the code in a program. Rubric used.
6.	Debugging: Demonstrate the use of a debugging tool in at least two Integrated Development Environments, with at least two languages.	Every semester of every year: 2016-2017	Test given as a part of the Capstone Class	Debugging: Demonstrate the use of a debugging tool in at least two Integrated Development Environments, with at least two languages. Rubric used.
7.	Linux: Students will demonstrate how to install, configure, create user accounts, issue correct commands and options, and perform standard network administration.	Every semester of every year: 2015-2017	Course-wide evaluation using a Linux Project measured using a common rubric.	Linux: Students will demonstrate how to install, configure, create user accounts, issue correct commands and options, and perform standard network administration.
8.				
9.				
10.				