CNM ANNUAL STUDENT LEARNING ASSESSMENT REPORT

Due to the Student Academic Assessment Committee by October 15

Central New Mexico Community College

PART 1: REPORT INFORMATION

Report Year and Contact Information			
2017-2018	Ivonne Nelson	inelson1@cnm.edu	50270
Academic Year	**Contact Person**	**CNM Email**	**CNM Office Extension**

Subject of this Report
BIT--CIS_AAS--CIS Computer Programming Concentratio

PART 2: CONTEXT IN WHICH THE ASSESSMENT TOOK PLACE

Program/Area Highlights and Successes (Wherever applicable, include course completion rates, job placement outcomes, and licensing examination pass rates. See the program information dashboard at https://livecnm.sharepoint.com/sites/Dashboards/SitePages/Program%20Information%20Dashboard.aspx (access restricted to CNM employees) and other reports at https://www.cnm.edu/depts/opie.)

Our students achieved targets in all assessment areas. Most notably, the debugging score has increased from 3.04 average to 3.29 average.

Changes Implemented During the Past Year in Support of Student Learning

Increased attention to debugging skills. 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.

CIS Programming instructors continue to keep curriculum up-to-date with quickly changing technology. We now teach two GUI frameworks in Java and C# because of changes in the industry. In C# .NET Web Development, Microsoft implemented a major change in how web development is accomplished and Android changed its template for implementing lists, driving a major rewrite of both curricula. All of the languages taught evolve and new additions/changes are incorporated into the curriculum as they are incorporated into the IDE's that are used.

PART 3: REPORT ON ASSESSMENT OF STUDENT LEARNING

Assessment Method	**Type of Assessment Tool**	**Population or Course(s) Assessed**	**Graduate Learning Outcome(s) Assessed**	**Mastery Level** (E.g., "Minimum score of 3 on a rubric scaled 0-4" or "Minimum score of 75%")	**Targeted % Achieving Mastery**	**Outcome**
Program Portfolio Demonstration	Direct & Internal	All CIS Computer Programming students were assessed in their final semester via the CIS 2999 Capstone course.	1. Class construction: Write programs that contain a programmer-written class and demonstrate its use in the C++, Java and C# languages.	The Computer Programming exit competencies are evaluated using a Rubrics with a scale of 4=excellent, 3=good, 2=fair and 1=poor.	92%	Target met
Program Portfolio Demonstration	Direct & Internal	All 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.	The Computer Programming exit competencies are evaluated using a Rubrics with a scale of 4=excellent, 3=good, 2=fair and 1=poor.	88%	Target met

Program Portfolio Demonstration	Direct & Internal	All CIS Computer Programming students were assessed in their final semester via the CIS 2999 Capstone course.	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.	The Computer Programming exit competencies are evaluated using a Rubrics with a scale of 4=excellent, 3=good, 2=fair and 1=poor.	96%	Target met
Program Portfolio Demonstration	Direct & Internal	All CIS Computer Programming students were assessed in their final semester via the CIS 2999 Capstone course.	4. Database manipulation and Web Application: Write a program that demonstrates the ability to connect to and manipulate a SQL database.	The Computer Programming exit competencies are evaluated using a Rubrics with a scale of 4=excellent, 3=good, 2=fair and 1=poor.	96%	Target met
Program Portfolio Demonstration	Direct & Internal	All CIS Computer Programming students were assessed in their final semester via the CIS 2999 Capstone course.	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.	The Computer Programming exit competencies are evaluated using a Rubrics with a scale of 4=excellent, 3=good, 2=fair and 1=poor.	96%	Target met

Test given as a part of the Capstone Class	Direct & Internal	All CIS Computer Programming students were assessed in their final semester via the CIS 2999 Capstone course.	6. Debugging: Demonstrate the use of a debugging tool in at least two Integrated Development Environments, with at least two languages.	The Computer Programming exit competencies are evaluated using a Rubrics with a scale of 4=excellent, 3=good, 2=fair and 1=poor.	84%	Target met
Course-wide evaluation using a Linux Project measured using a common rubric.	Direct & Internal	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.	7. Linux: Students will demonstrate how to install, configure, create user accounts, issue correct commands and options, and perform standard network administration.	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.	80%	N/A

Summary of Assessment Findings

	COMP 1	COMP 2	COMP 3	COMP 4	COMP 5	COMP 6
SCORE	**CLASSES**	**INHERITANCE**	**GUI**	**DB MANIP**	**RESEARCH**	**DEBUG**
4	12	14	5	15	15	9
3.5 – 3.9	7	5	11	4	6	7
3	4	3	8	5	3	5

2.5	1	1	1	1		3
2	1	2				1
1.5						
1						
0					1	
	COMP 1	**COMP 2**	**COMP 3**	**COMP 4**	**COMP 5**	**COMP 6**
SCORE	**CLASSES**	**INHERITANCE**	**GUI**	**DB MANIP**	**RESEARCH**	**DEBUG**
3+	23	22	24	24	24	21
<3	2	3	1	1	1	4
Average	3.68	3.52	3.84	3.84	3.84	3.36
Meet Target?	yes	yes	yes	yes	yes	yes

Interpretation of Assessment Findings

Although all of the targets were met, the Debugging target was the lowest. The next lowest was Inheritance and polymorphism. The remaining areas averaged out to be 95%, while debugging was at 84 percent and inheritance and polymorphism was at 88%. The Debugging skill is still the one that needs the most improvement.

Action Plan in Support of Student Learning (Describe changes to be made that are based at least in part on the assessment interpretation. If the assessment did not yield useful information, describe changes to be made in the assessment methodology and/or criteria.)

The three Programming full time faculty and any current part-time faculty will meet to discuss strategies to achieve an improvement in debugging skills. We are open to ideas in many areas, including those checked below.

Please select all of the following that characterize the types of changes described in the above action plan:

- ☐ Assessment criteria revision
- ☐ Assessment methodology revision
- ☒ Assignment revision
- ☐ Budgetary reallocation
- ☒ Change in teaching approach
- ☐ Course content revision
- ☒ Curricular Revision
- ☒ Faculty training/development
- ☐ Process revision

Recommendations, Proposals, and/or Funding Requests	Budget Needed

PART 4: REMAINING YEARS IN CURRENT ASSESSMENT CYCLE PLAN (including any revisions) **– OR -- UPCOMING ASSESSMENT CYCLE PLAN** (if this was the final year)

Years of Full Cycle	**Next Year's Assessment Focus** (Describe how the next planned assessment is expected to provide information that can be used toward improving student learning.)
2016-2022	No Changes Made

Graduate Learning Outcomes to Be Assessed	Years in which Assessment Is Planned	Population/Courses to Be Assessed	Planned Assessment Approach
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: 2017-2018	Cli Program Portfolio Demonstration ck or tap here to enter text.	Class construction: Write programs that contain a programmer-written class and demonstrate its use in the C++, Java and C# languages. Rubric used.
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: 2017-2018	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.
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: 2017-2018	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.
Database manipulation and Web Application: Write a program that demonstrates the ability to connect to and manipulate a SQL database.	Every semester of every year: 2017-2018	Program Portfolio Demonstration	Database manipulation and Web Application: Write a program that demonstrates the ability to connect to and manipulate a SQL database. Rubric used.
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: 2017-2018	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.

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: 2017-2018	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.
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: 2017-2018	Course-wide evaluation using a Linux Project measured using a common rubric.	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.