Course Title: Foundations of Programming
Course Number: 9007210

NGSSS Benchmark	Content Focus	Number of Questions	Suggested Cognitive Complexity *(per CPALMS)*
Reporting Category 31: **31.0 Explore the characteristics, tasks, work attributes, options, and tools associated with a career in software development.**			
31.06	31.06 Explain the need for continuing education and training of computer programmers.	1	1 level 2
31.08	31.08 Describe ethical responsibilities of computer programmers.	1	1 level 2
31.11	31.11 Identify devices, tools, and other environments for which programmers may develop software.	1	1 level 1
Reporting Category Total		3	
Reporting Category 32: **32.0 Demonstrate an understanding of the characteristics, use, and selection of numerical, non-numerical, and logical data types.**			
32.02	32.02 Explain the types and uses of variables in programs.	3	1 level 1 2 level 2
32.03	32.03 Determine the best data type to use for given programming problems.	2	1 level 2 1 level 3
32.08	32.08 Use different number systems to represent data.	1	1 level 2
32.10	32.10 Use Boolean logic to perform logical operations.	2	1 level 2 1 level 3
Reporting Category Total		8	
Reporting Category 33: **33.0 Distinguish between iterative and non-iterative program control structures.**			
33.01	33.01 Explain non-iterative programming structures (*e.g.*, if, if/else) and their uses.	2	2 level 2
33.02	33.02 Explain iterative programming structures (*e.g.*, while, do/while) and their uses.	1	1 level 1
Reporting Category Total		3	
Reporting Category 35: **35.0 Describe the processes, methods, and conventions for software development and maintenance.**			
35.04	35.04 List and explain the steps in the program development cycle.	2	1 level 2 1 level 3
35.06	35.06 Describe the on-going need for program maintenance.	1	1 level 2
Reporting Category Total		3	
Reporting Category 36: **36.0 Explain the types, uses, and limitations of testing for ensuring quality control.**			
36.02	36.02 Explain testing performed at different stages of the program development cycle (*e.g.* unit testing, system testing, user acceptance testing).	1	1 level 2
36.04	36.04 Describe and identify types of programming errors (*e.g.*, syntactical, logic, usability, requirements mismatch).	2	1 level 1 1 level 3
Reporting Category Total		3	
Reporting Category 37: **37.0 Create a program design document using Unified Modeling Language**			

(UML) or other common design tool.			
37.02	37.02 Describe tools for developing a program design (*e.g.*, UML, flowcharts, design documents, pseudocode).	1	1 level 2
37.03	37.03 Explain the role of existing libraries and packages in facilitating programmer productivity.	1	1 level 2
Reporting Category Total		2	
Reporting Category 38: **38.0 Describe information security risks, threats, and strategies associated with software development.**			
38.02	38.02 Identify different types of threats to computer systems.	1	1 level 1
Reporting Category Total		1	
Reporting Category 41: **41.0 Use information technology tools.**			
41.02	41.02 Employ technological tools to expedite workflow including word processing, databases, reports, spreadsheets, multimedia presentations, electronic calendar, contacts, email, and internet applications.	1	1 level 2
41.04	41.04 Employ collaborative/groupware applications to facilitate group work.	1	1 level 2
Reporting Category Total		2	

Overall Percentage for Written Test:__70%_____
Overall Percentage for Performance Tasks:__30%_____

Course Title: Foundations of Programming
Course Number: 9007210

Overall Percentage for Performance Tasks:__30%_____

Performance Task #1	**Python Survival Game**
Weighting Percent for this Task	30%
Standard	32.0 Demonstrate an understanding of the characteristics, use, and selection of numerical, non-numerical, and logical data types. 33.0 Distinguish between iterative and non-iterative program control structures. 36.0 Explain the types, uses, and limitations of testing for ensuring quality control. 39.0 Use oral and written communication skills in creating, expressing and interpreting information and ideas. 40.0 Solve problems using critical thinking skills, creativity and innovation. 41.0 Use information technology tools.
Exemplar	

(If applicable)	
Additional Information	Students will individually build a text-based survival game using the Python programming language.
Suggested Assessment Team	

Rubric:

Foundations of Programming EOC Performance Task - Assessment List

Domain	Element	Points Possible	Earned Assessment	
			Student	Teacher
32: Data types	Correctly uses variables with at least 3 data types (integer, float, string, Boolean)	10		
	Correctly uses arithmetic and logical expressions	10		
33: Iterative and non-iterative	Correctly uses iterative structures (if/else statements)	10		
	Correctly uses non-iterative structures (while loops)	10		
36: Testing	Game is free from syntactical, logic, usability errors	15		
39: Information and ideas	Design, develop, and deliver an informal presentation to seek feedback	5		
	Design, develop, and deliver a formal presentation to engage and inform	5		
	Gameplay flows well and makes sense to the user	10		
40: Solving Problems	Create benchmark goals in order to submit on time	5		
	Employ critical thinking skills to ensure that the game is free from errors or problems	5		
	The user is required to make a series of at least 5 choices that lead to a clear win or loss	10		

41: Tech tools	Use a variety of technology tools to manage workflow and store information	5		
	Totals			
Notes				

To Do:
Create 25 test questions.
Who grades my performance task?