

COMP B11: Programming Concepts and Methodology I

Student Learning Outcomes	Measure	PLO	ILO	GE
<p>1. Understand and know how to use fundamental programming constructs</p> <ul style="list-style-type: none"> • Analyze and explain the behavior of simple programs involving the fundamental programming constructs covered by this unit. • Modify and expand short programs that use standard conditional and iterative control structures and functions. • Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions. • Choose appropriate conditional and iteration constructs for a given programming task. • Apply the techniques of structured (functional) decomposition to break a program into smaller pieces. • Describe the mechanics of parameter passing. • Design, implement, test, and debug simple programs in an object-oriented programming language. 	<ul style="list-style-type: none"> • Written exams 	1	I	
	<ul style="list-style-type: none"> • Programming Exams • Programming Assignments • In-class Pairs Programming • Quizzes 	3	III	
<p>2. Design and implement algorithms to solve problems</p> <ul style="list-style-type: none"> • Discuss the importance of algorithms in the problem-solving process. • Identify the necessary properties of good algorithms. • Create algorithms for solving simple problems. • Use pseudocode or a programming language to implement, test, and debug algorithms for solving simple problems. • Describe strategies that are useful in debugging. 	<ul style="list-style-type: none"> • Written exams 	1	I	
	<ul style="list-style-type: none"> • Programming Exams • Programming Assignments • In-class Pairs Programming • Quizzes 	3	III	
<p>3. Discuss general programming-language ideas</p>	<ul style="list-style-type: none"> • Written exams 	1	I	
	<ul style="list-style-type: none"> • Programming Exams 	3	III	

- Summarize the evolution of programming languages illustrating how this history has led to the paradigms available today.
 - Identify at least one distinguishing characteristic for each of the programming paradigms covered in this unit.
4. Understand and use variables and types
- Explain the value of declaration models, especially with respect to programming-in-the-large.
 - Identify and describe the properties of a variable such as its associated address, value, scope, persistence, and size.
 - Discuss type incompatibility.
 - Demonstrate different forms of binding, visibility, scoping, and lifetime management.
 - Defend the importance of types and type-checking in providing abstraction and safety.
 - Design and implement user-defined types, using Object-Oriented principles.

<ul style="list-style-type: none"> • Programming Assignments • In-class Pairs Programming • Quizzes 			
<ul style="list-style-type: none"> • Written exams • Programming Exams • Programming Assignments • In-class Pairs Programming • Quizzes 	1	I	
	3	III	

PLOs:

1. 1. Identify the appropriate software development technologies, algorithms, and scientific and mathematical principles to apply to a given problem.

Assessment: The department will evaluate using examinations.

2. Effectively design and implement programming constructs, including functions, control structures, arrays/lists, classes and objects for a given programming problem.

Assessment: The department will evaluate using examinations.

3. Effectively implement the appropriate data structures using the principles and techniques of object-oriented programming for a given programming problem.

Assessment: The department will evaluate using examinations.

ILOs:

- I. Think critically and evaluate sources and information for validity and usefulness.
- II. Communicate effectively in both written and oral forms.
- III. Demonstrate competency in a field of knowledge or with job-related skills.

IV. Engage productively in all levels of society – interpersonal, community, the state and nation, and the world.