

		Course Code: CPT 122		Credit Hours: 4
	Subject/Course: Programming Principles I			Theoretical: 1 hours/week
	Year: One	Semester: Two	Pre-requisite: CPT 112	Practical: 3 hours /week

General Objectives This course is designed to provide a comprehensive introduction to programming, and builds a solid foundation of programming skills that can be used to master additional programming languages like C, C++, or Java. In this course you will write, compile, and debug efficiently, reliable programs in High level languages

- 1- Introduction to programming
- 2- Introduction to High level language programs e.g. C Language
- 3- Understand Data Types and operators
- 4- Understand Object Oriented Concepts and Programming.
- 5- Understand Classes and Objects
- 6- Understand Control Statements
- 7- Understand Method (Sub-Routine and Function)
- 8- Understand the Use and Application of Arrays
- 9- Know the Concept of Exception Handling (Debugging)
- 10- Understand Working with Strings and Characters
- 11- Understand Working with Files and Streams

	Course: Programming Principles I		Course Code: CPT 122			Credit Hours: 4
						Theoretical: 1 hours/week
	Year: One	Semester: One	Pre-requisite: CPT 112			Practical: 3 hours/week
Theoretical Content				Practical Content		
General Objective 1: Introduction to programming						
Week	Specific Learning Outcomes	Teacher's activities	Resources	Specific Learning Outcomes	Teacher's activities	Resources
1	Understand basic concepts and definitions of programming.	<p>Explain the basic concepts of programming:</p> <ul style="list-style-type: none"> • Define what a program is. • Give example of a simple program e.g. "Hello World!" • Explain Programming Process • Explain Program Files and Program Execution • Differentiate between System Programs and Application Programs • Explain Input – Process – Output • Explain the meaning of Programming Language and give examples of different programming Languages • Explain Programming Errors and differentiate between: Compile Error and Runtime Error. • Explain programming Development Environments 	<ul style="list-style-type: none"> • Computer System • A comprehensive workbook of Programming Fundamentals and principles 	Develop a simple program that prints the message, "Hello World!" in the Console Environment.	Supervise the laboratory and support students in their practical work	<ul style="list-style-type: none"> • Computer System loaded with High level programming language compilers • A comprehensive workbook for students
General Objective 2: Introduction to High level language programs e.g. Java Programming Language						
2	<ul style="list-style-type: none"> • Know the process of writing Simple Programs • Know how to use input and 	<p>Explain the steps of writing a simple program including :</p> <ul style="list-style-type: none"> - Reading Input - Performing Numeric Calculations - Formatting Output - Decision Making 	<ul style="list-style-type: none"> • Computer System • A comprehensive workbook of Programming Fundamentals 	<p>Write a simple program that includes :</p> <ul style="list-style-type: none"> - Reading Input - Performing Numeric Calculations 	Supervise the laboratory and support students in their practical work	<ul style="list-style-type: none"> • Computer System loaded with High level programming

	<p>output statements</p> <ul style="list-style-type: none"> • Use Arithmetic Operators • Know the precedence of Arithmetic Operators • Know basic memory concepts 	<ul style="list-style-type: none"> - Iteration - Commenting Source Code - Good Programming Style - Explain basic memory concepts. 	<p>and principles</p>	<ul style="list-style-type: none"> - Formatting Output - Commenting Source Code - Good Programming Style 		<p>ng language compliers</p> <ul style="list-style-type: none"> • A comprehensive workbook for students
General Objective 3: Understand Data Types and operators						
3	<ul style="list-style-type: none"> • Know Data Types. • Know the difference between Constants and Variables. • Know the difference between primitive data types and reference data types. • Know how to write decision making statements. • Use relational and equality operators 	<ul style="list-style-type: none"> - Explain Data Types, Constants, and Variables - Describe Program's Purpose as the ability to Process Data - Give examples of Different Data Types - Describe decision making statements. - Explain primitive and reference data types 	<ul style="list-style-type: none"> • Computer System • A comprehensive workbook of Programming Fundamentals and principles 	<p>Write a program that declares and uses different data types which include simple decision making statement and uses relational and equality operators.</p>	<p>Supervise the laboratory and support students in their practical work</p>	<ul style="list-style-type: none"> • Computer System loaded with High level programming language compliers • A comprehensive workbook for students
General Objective 4: : Understand Object Oriented Concepts and Programming						

4	<ul style="list-style-type: none"> Understand Basic Computer-technology concepts, such as classes, objects, attributes, behaviours, encapsulation, inheritance and polymorphism. 	<p>Explain basic computer technology concepts; such as classes, objects, attributes, behaviours, encapsulation, inheritance and polymorphism.</p>	<ul style="list-style-type: none"> Computer System A comprehensive workbook of Programming Fundamentals and principles 	<p>Use Classes from existing API's and Libraries and create objects from these classes.</p>	<p>Supervise the laboratory and support students in their practical work</p>	
General Objective 5: Understand Classes and Objects						
5	<ul style="list-style-type: none"> Know what classes, objects, methods and instance variables are. Know how to use basic problem solving techniques Know how to develop algorithms through the process of top-down, stepwise refinement using pseudocode 	<ul style="list-style-type: none"> Explain the declaration of classes Explain concrete class and abstract class Explain the instantiation of objects from classes. Explain instance variables Explain the process of basic problem solving techniques. Explain algorithm and pseudocode. Formulate simple algorithm through the process of top-down, stepwise refinement using pseudocode. 	<ul style="list-style-type: none"> Computer System A comprehensive workbook of Programming Fundamentals and principles 	<ul style="list-style-type: none"> Write a program that defines and declares a class Create objects from the class defined and use these objects in another class. 	<p>Supervise the laboratory and support students in their practical work</p>	<ul style="list-style-type: none"> Computer System loaded with High level programming language compilers A comprehensive workbook for students
General Objective 6: Understand Control Statements						
6	<ul style="list-style-type: none"> Know how to use the "if" and "if...else" selection 	<ul style="list-style-type: none"> Explain the if single-selection statement. Explain the if...else double-selection statement. 	<ul style="list-style-type: none"> Computer System A comprehensive 	<ul style="list-style-type: none"> Write a simple program that shows the use of "if" and "if...else" selection 	<p>Supervise the laboratory and support students in their practical</p>	<ul style="list-style-type: none"> Computer System loaded with High

	<p>statements to choose among alternative actions.</p> <ul style="list-style-type: none"> • Know how to use the “while” repetition statement to execute statements in a program repeatedly • Know how to use counter-controlled repetition and sentinel-controlled repetition. • Know how to use the compound assignment, increment and decrement operators. 	<ul style="list-style-type: none"> • Explain the while repetition statement. • Explain the construction of counter-controlled repetition using the while repetition statement. • Explain the construction of sentinel-controlled repetition using the while repetition statement. • Explain compound statement. • Explain increment and decrement operators. 	<p>workbook of Programming Fundamentals and principles</p>	<p>statements.</p> <ul style="list-style-type: none"> • Write a simple program that shows the use of “while” repetition statement showing the use of counter controlled repetition and sentinel-controlled repetition. • Write a simple program to shows the use of compound assignment, increment and decrement operators. 	<p>work</p>	<p>level programming language compliers</p> <ul style="list-style-type: none"> • A comprehensive workbook for students
7	<ul style="list-style-type: none"> • Know the use of “for” and “do...while” repetition statements to execute statements in a program repeatedly. • Understand multiple 	<ul style="list-style-type: none"> • Describe the use of “for” and “do...while” repetition statements to execute statements in a program repeatedly. • Describe the use of multiple selection using “switch” selection statement in place of multiple if...else selection statement. • Explain how to use logical operators to form complex conditional expressions in control statements. 	<ul style="list-style-type: none"> • Computer System • A comprehensive workbook of Programming Fundamentals and principles 	<ul style="list-style-type: none"> • Write a simple program that shows the use of “for” and “do...while” repetition statements. • Write a simple program that shows the use of “switch” statement to make multiple selection. • Write a simple 	<p>Supervise the laboratory and support students in their practical work</p>	<ul style="list-style-type: none"> • Computer System loaded with High level programming language compliers • A comprehen

	<p>selection using the “switch” selection statement</p> <ul style="list-style-type: none"> Know the use of logical operators to form complex conditional expressions in control statements 			<p>program that uses logical operators to form complex conditional expressions in control statements.</p>		<p>sive workbook for students</p>
General Objective 7: Understand Method (Sub-Routine and Function)						
Week	Specific Learning Outcomes	Teacher’s activities	Resources	Specific Learning Outcomes	Teacher’s activities	Resources
8	<ul style="list-style-type: none"> Know the concepts of Method (Subroutines and functions) Know the scope of variables Know the mechanisms for passing information between methods. 	<ul style="list-style-type: none"> Explain what methods are. Explain how methods are declared. Explain Method return a value (Function) Explain non-returning value Method (Sub-routine) Explain the scope of variables Use Methods from standard API’s and Libraries Explain the mechanism for passing information between methods. 	<ul style="list-style-type: none"> Computer System A comprehensive workbook of Programming Fundamentals and principles 	<p>Write a simple class that declares and uses simple methods.</p>	<p>Supervise the laboratory and support students in their practical work</p>	<ul style="list-style-type: none"> Computer System loaded with High level programming language compilers A comprehensive workbook for students
General Objective 8: Understand the Use and Application of Arrays						
9	<ul style="list-style-type: none"> Know what arrays are. Know how to declare arrays, initialize arrays 	<ul style="list-style-type: none"> Explain arrays. Declare and initialize arrays Explain reference to individual array elements. Describe how to use arrays to store 	<ul style="list-style-type: none"> Computer System A comprehensive workbook of Programming 	<p>Write a simple program that declares and initializes one dimensional array and assesses the array data.</p>	<p>Supervise the laboratory and support students in their practical work</p>	<ul style="list-style-type: none"> Computer System loaded with High level programmi

	<p>and refer to individual elements of arrays.</p> <ul style="list-style-type: none"> • Know how to use array to store data in and retrieve data from lists. 	<p>data and retrieve data from lists.</p>	<p>Fundamentals and principles</p>			<p>ng language compliers</p> <ul style="list-style-type: none"> • A comprehensive workbook for students
10	<ul style="list-style-type: none"> • Know Multi-Dimensional arrays. • Know how to declare, initialize and refer to individual elements of Multi-Dimensional arrays. • Know how to use Multi-Dimensional arrays to store and retrieve data from tables of values. 	<ul style="list-style-type: none"> • Explain Multi-Dimensional arrays. • Declare and initialize Multi-Dimensional arrays • Explain reference to individual Multi-Dimensional array elements. • Describe how to use Multi-Dimensional arrays to store data and retrieve data from tables of values. 	<ul style="list-style-type: none"> • Computer System • A comprehensive workbook of Programming Fundamentals and principles 	<p>Write a simple program that declares and initializes multi-dimensional array and assesses the array data.</p>	<p>Supervise the laboratory and support students in their practical work</p>	<ul style="list-style-type: none"> • Computer System loaded with High level programming language compliers • A comprehensive workbook for students
General Objective 9: : Know the Concepts of Exception Handling (Debugging)						
11	<ul style="list-style-type: none"> • Understand the concepts of Exceptions and Exceptions Handling. • Know how Exception and 	<ul style="list-style-type: none"> • Describe the concepts of Exceptions and Exceptions Handling. • Describe the working of exception and error handling in a program. • Explain common exceptions e.g. arithmetic exception, number format exception, input-output exception. 	<ul style="list-style-type: none"> • Computer System • A comprehensive workbook of Programming Fundamentals and principles 	<p>Write a program that throws exceptions and handle the exceptions.</p>	<p>Supervise the laboratory and support students in their practical work</p>	<ul style="list-style-type: none"> • Computer System loaded with High level programming language

	<p>error handling works.</p> <ul style="list-style-type: none"> Understand the use of “try”, “throw” and “catch” to detect, indicate and handle exception. Understand the use of “finally block” to release resources. 	<ul style="list-style-type: none"> Describe the use of try, throw and catch, to detect, indicate and handle exceptions. Describe the use of finally block to release resources. 				<p>compilers</p> <ul style="list-style-type: none"> A comprehensive workbook for students
12	<ul style="list-style-type: none"> Know how to declare new exception classes. Know how stack unwinding enables exceptions not caught in one scope to be caught in another. Know how stack traces help in debugging. Know how to create chained exceptions that maintain complete stack-trace information. 	<ul style="list-style-type: none"> Explain the creation of new exception classes. Explain how stack unwinding enables exceptions not caught in one scope to be caught in another. Explain how stack traces help in debugging. Explain the creation of chained exceptions that maintain complete stack-trace information. 	<ul style="list-style-type: none"> Computer System A comprehensive workbook of Programming Fundamentals and principles 	<ul style="list-style-type: none"> Write a program that defines a new exception class Write another program that throws the exception and handle the same exception. 	Supervise the laboratory and support students in their practical work	<ul style="list-style-type: none"> Computer System loaded with High level programming language compilers A comprehensive workbook for students
<p>General Objective 10: : Understand Working with Strings and Characters</p>						

13	<ul style="list-style-type: none"> • Know how to create and manipulate immutable character-string objects. • Know how to create and manipulate mutable character-string objects • Know how to create and manipulate character objects • Know how to break strings into tokens. 	<ul style="list-style-type: none"> • Explain immutable strings and demonstrate how to create and manipulate immutable strings. • Explain mutable strings and demonstrate how to create and manipulate mutable strings. • Explain character objects and demonstrate how to create and manipulate character objects. • Show how to break strings into tokens. 	<ul style="list-style-type: none"> • Computer System • A comprehensive workbook of Programming Fundamentals and principles 	Write a program that manipulates characters, mutable and immutable strings.	Supervise the laboratory and support students in their practical work	<ul style="list-style-type: none"> • Computer System loaded with High level programming language compilers • A comprehensive workbook for students
General Objective 11: Understand working with Files and Streams						
14	<ul style="list-style-type: none"> • Know what files are and how files are used to retain application data between successive executions. • Know how to create, read, write and update files • Know how to retrieve information about files and directories 	<ul style="list-style-type: none"> • Explain what files are and how files are used to retain application data between successive executions. • Explain the process of creating, reading, writing and updating files. • Describe the process of retrieving information about files and directories 	<ul style="list-style-type: none"> • Computer System • A comprehensive workbook of Programming Fundamentals and principles 	Write a program that creates file objects, that reads, writes and update the files, which also retrieves information about the files and directories.	Supervise the laboratory and support students in their practical work	<ul style="list-style-type: none"> • Computer System loaded with High level programming language compilers • A comprehensive workbook for students

15	<ul style="list-style-type: none"> • Know input/output streams class hierarchy. • Know the difference between text files and binary files. • Know sequential-access file processing • Know how to use file input stream and file output stream to read and write to files. 	<ul style="list-style-type: none"> • Explain basic Input/Output streams class hierarchy. • Explain the difference between text files and binary files. • Explain sequential-access file processing. • Describe how to use file input stream and file output stream to read and write to files. 	<ul style="list-style-type: none"> • Computer System • A comprehensive workbook of Programming Fundamentals and principles 	Write a program that create and uses binary and text sequential-access files.	Supervise the laboratory and support students in their practical work	<ul style="list-style-type: none"> • Computer System loaded with High level programming language compliers • A comprehensive workbook for students
----	--	--	--	---	---	--

Assessment: Give details of assignments to be used:
 Coursework/ Assignments %; Course test %; Practical %; Projects %; Examination %

Recommended Textbooks & References: