Module: Programming Preliminaries 161

Module name:	Programming Preliminaries 161
Code:	PRL161
NQF level:	5
Type:	Core – Diploma in Information Technology (all stream)
Contact time:	72 hours
Structured time:	12 hours
Self-directed time:	76 hours
Notional hours:	160 hours
Credits:	16
Prerequisites:	None

Purpose

The purpose of this module is to understand how software and the program has helped people solve problems. The student will learn several general concepts that will allow them to formulate an understanding of a problem and develop a process and algorithm to support the solution.

Outcomes

Upon successful completion of this module, the student will be able to:

- Demonstrate the ability to identify, evaluate and solve defined, routine and new problems within a familiar context.
- Demonstrate a fundamental understanding of problem-solving concepts, their definition, scope, and the underlying principles including critical and computational thinking, and metacognition.
- Demonstrate a fundamental understanding of problem-solving using algorithms.
- Demonstrate the ability to apply solutions based on relevant evidence and procedures or other forms of explanation appropriate to the field, discipline or practice, demonstrating an understanding of the consequences.
- Identify and evaluate the problem context and, through applying various criteria, select the optimal problem-solving method.
- Identify, evaluate and solve defined arithmetic problems, apply solutions based on relevant evidence and understand the consequences if an algorithm is poorly designed.
- Select and apply standard problem-solving techniques, and plan and manage an implementation process applied to problem-solving.
- Select and apply solutions based on relevant evidence and procedures or other forms of explanation appropriate to the field, discipline or practise, demonstrating an understanding of the consequences.
- Operate in a range of familiar problem domains, demonstrating an understanding of different kinds of problems to be solved, their constituent parts and the relationships between these parts, and to understand how algorithms in one area impact other areas within the same domain.

Assessment

- Continuous evaluation of theoretical work through three written assignments, one project, two formative tests, and one summative test.
- Continuous evaluation through tracking of progress, offering support, guidance and provision of constant stream of opportunities to prove mastery of subject material and pursuing more challenging work as they master the basics.
- Final assessment through an examination.
- The assignments or projects collectively will count 20% of your class mark.
- All tests will collectively account for 80% of your class mark.
- Your class mark contributes 30% towards your final mark for the subject, while the final assessment accounts for 70% of your final mark.

Teaching and Learning

Learning materials

- 🕮 ***A., B. (2016)*** **Problem Solving and Computer Programming Using C.** ***New Delhi: Laxmi Publications Pvt Ltd.***
- 🕮 ***Karl Beecher (2017)*** **Computational Thinking : A Beginner's Guide to Problem-solving and Programming.** ***Swindon, UK: BCS, The Chartered Institute for IT.***
- 🕮 ***Tripathi, A. K. and Tripathi, M. (2015)*** **Computer System and Programming in 'C'.** ***New Delhi, India: Laxmi Publications Pvt Ltd.***
- 🕮 ***Srivastava, A. K. (2020)*** **A Practical Approach to Data Structure and Algorithm with Programming in C.** ***Oakville, ON: Arcler Press.***

Learning activities

Learning will be facilitated by the lecturer with student centred activities that involve problem-based learning where pupils are presented with challenges that replicate the situation in the real-world environment. This will be achieved through a combination between presentation of theoretical concepts, guided exercises, group work and discussions during the module.

Notional learning hours

Activity	Units	Contact Time	Structured Time	Self-Directed Time
Lecture		60.0		39.0
Formative feedback		12.0		
Project	1			12.0
Assignment	3			6.0
Test	3		9.0	12.0
Exam	1		3.0	7.0
		72.0	**12.0**	**76.0**

Syllabus

- Problem Definitions
- Problem Solving Techniques
- Problem Solving Approaches
- Logical Reasoning
- Critical Thinking

- Computational Thinking
- Metacognition
- Introduction to Programming
- Programming Techniques
- Variables and Expressions
- Decision and Loop Control Statements
- Logic Representation
    - Pseudocode
    - Algorithm
    - Flowchart