Module: Programming 161

Module name:	Programming 161
Code:	PRG161
NQF level:	5
Type:	Core – Diploma in Information Technology (all stream)
Contact time:	34 hours
Structured time:	6 hours
Self-directed time:	40 hours
Notional hours:	120 hours
Credits:	12
Prerequisites:	Programming Principles 161; Mathematics 161

Purpose

This course aims to expand on the mastered knowledge obtained from Programming Principles. This module will broaden the students basic programming skills by looking into variables, methods, condition checking and decision making.

Outcomes

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

- Identify, evaluate and solve defined, routine problems within the programming context, and apply optimal solutions based on relevant evidence and procedures or other forms of explanation.
- Demonstrate an informed understanding of variables, expressions, decision making, condition checking.
- Demonstrate an informed understanding of the key terms, concepts, general principles, rules and theories in the programming domains.
- Select and apply standard rules, procedures or techniques to solve a problem within the programming domains.
- Gather information from a range of sources, select information appropriate to the task, and apply basic processes of analysis, synthesis and evaluation on that information to determine its suitability as an optimal solution.
- Demonstrate the ability to communicate information reliably, accurately and coherently, using conventions appropriate to the context.
- Effectively implement a sorting and searching algorithm based on a given datastructure.

Assessment

- Continuous evaluation of theoretical work through written assignment, a formative, and a summative test.
- Continuous evaluation of project work, where the student must design, manage and report on the evaluation of testing methodologies and the selection of an appropriate methodology for a given scenario, justifying the choice made with well-formed arguments and evidence.
- Final assessment through a written examination.
- The assignments or projects collectively will count 30% of your class mark.

- All tests will collectively account for 70% 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

- ***Anandamurugan, S. (2014)*** **C Programming for Juniors*.*** ***New York: Nova Science Publishers, Inc (Computer Science, Technology and Applications).***
- ***A., B. (2016)*** **Problem Solving and Computer Programming Using C*.*** ***New Delhi: Laxmi Publications Pvt Ltd.***
- ***Yadav, A. K. and Yadav, V. K. (2019)*** **Data Structures with C Programming*.*** ***Ashland: Arcler Press.***
- ***Srivastava, A. K. (2020)*** **A Practical Approach to Data Structure and Algorithm with Programming in C*.*** ***Oakville, ON: Arcler Press***

Learning activities

The teaching and learning activities are an amalgamation of formal lectures based around theoretical concepts and practical guided instructions where the actual implementation can be practised. The practical components also serve to indicate areas of discussion that are needed to further flesh out some concepts. One mandatory assignment and one project must be completed during the course. The experiences and progress on these practical components form the content of class discussions.

Notional learning hours

Activity	Units	Contact Time	Structured Time	Self-Directed Time
Lecture		27.0		13.0
Formative feedback		3.5		
Project	1	3.5		9.0
Assignment	1			3.0
Test	2		4.0	8.0
Exam	1		2.0	7.0
		34.0	**6.0**	**40.0**

Syllabus

- Programming process
- Introduction to Scratch Environment
- Input/Output
- Semantic vs Syntax
- Variables
- Expressions and Operators
- Decision and Loop Control Statements
- Sorting and Searching; Algorithms
- Datastructures: Array (1D), List