CS2315: COMPUTER PROGRAMMING

Effective Term
Semester A 2024/25

Part I Course Overview

Course Title
Computer Programming

Subject Code
CS - Computer Science
Course Number
2315

Academic Unit
Computer Science (CS)

College/School
College of Computing (CC)

Course Duration
One Semester

Credit Units
3

Level
B1, B2, B3, B4 - Bachelor's Degree

Medium of Instruction
English

Medium of Assessment
English

Prerequisites
Nil

Precursors
Nil

Equivalent Courses
CS1315 Introduction to Computer Programming
CS2310 Computer Programming
CS2311 Computer Programming

Exclusive Courses
CS2313 Computer Programming
CS2360 Java Programming

Part II Course Details

Abstract

This course aims to equip the students with key concepts and techniques of programming using a high-level object-oriented programming language and to develop practical skills in producing quality programs. No prior programming or computer science experience is required.

Course Intended Learning Outcomes (CILOs)

	CILOs	Weighting (if app.)	DEC-A1	DEC-A2	DEC-A3
1	Explain the structure of an object-oriented computer program.	10	x	x	
2	Analyze, test and debug computer programs.	15	x	x	
3	Solve a task by applying effective programming techniques, which involve algorithm and data structures.	60		x	
4	Design and construct well-structured programs with good programming practices.	15		x	x

A1: Attitude
Develop an attitude of discovery/innovation/creativity, as demonstrated by students possessing a strong sense of curiosity, asking questions actively, challenging assumptions or engaging in inquiry together with teachers.

A2: Ability
Develop the ability/skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas, acquiring research skills, synthesizing knowledge across disciplines or applying academic knowledge to real-life problems.

A3: Accomplishments
Demonstrate accomplishment of discovery/innovation/creativity through producing /constructing creative works/new artefacts, effective solutions to real-life problems or new processes.

Learning and Teaching Activities (LTAs)

	LTAs	Brief Description	CILO No.	Hours/week (if applicable)
1	Lecture	Students will engage with various programming concepts and techniques. The programming concepts will be explained and demonstrated with examples.	1, 2, 3, 4	3 hours per week

2	Lab	Students will put theory into practice and be proficient in a programming language. The laboratory exercises consist of programming tasks and students can try out their programs using a common integrated development environment. Feedback will be given to students on their work.	1, 2, 3, 4	1 hour per week
3	Assignment	Students will consider the given requirements of more comprehensive tasks and design programming solutions by applying and combining various techniques learnt from lectures and laboratory exercises. Students will implement their solutions as practical computer programs, and explain their ideas/ algorithms using suitable presentation methods (e.g. a report, flowchart, etc.).	2, 3, 4	After class

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks (e.g. Parameter for GenAI use)
1	Quiz	1, 3, 4	20	Correctly explain the structure of an object-oriented computer program
2	Assignment	2, 3, 4	20	Select proper test cases to assess the correctness of a program. Students are required to work on assignments at least once every four weeks

Continuous Assessment (%)

40

Examination (%)

60

Examination Duration (Hours)

2

Additional Information for ATs
For a student to pass the course, at least 30% of the maximum mark for the examination must be obtained.

Assessment Rubrics (AR)

Assessment Task
Quiz

Criterion
ABILITY to explain, analyse and debug the structure of a computer program

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Not even reaching marginal levels

Assessment Task
Assignment

Criterion
CAPACITY for applying programming techniques

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Not even reaching marginal levels

Assessment Task
Examination

Criterion
CAPACITY for analyzing and writing effective computer programs

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Not even reaching marginal levels

Part III Other Information

Keyword Syllabus

Program design, development of algorithms, programming language, control structures, data types, arrays, file I-O, recursion, and data structures, object-based programming: data abstraction, classes, and the class library; programming style, program testing.

Syllabus:

- Computers and programming
  Hardware/software hierarchy, the computer as a multi-level language machine. The software development process. Program development environments.
- Programming techniques and the development of algorithms
  Algorithms, programming language, modular decomposition and procedural abstraction, automatic and dynamic variables, parameter-passing by reference and by value for atomic data, objects, and arrays, control structures, iteration.
- Data structures
  The concept of data types. Simple data types. Arrays. Strings. Files. Data abstraction: encapsulation, information hiding. Defining and using classes. The class library. Data structures.
- Program development practice
  Professional programming styles. Program testing. Program documentation.

Reading List

Compulsory Readings

	Title
1	Richard L. Halterman (2015). Fundamentals of C++ Programming. Southern Adventist University.

Additional Readings

	Title
1	S.B. Lippman, J. Lajoie and B. Moo (2012). C++ Primer. Addison Wesley, 5th edition.
2	H.M. Deitel & P.J. Deitel (2011). C++ How to Program. Pearson Int. Edition, 8th edition.
3	Walter Savitc (2010). Absolute C++. Addison-Wesley, 4th edition.