CS2313: COMPUTER PROGRAMMING

Effective Term
Semester A 2023/24

Part I Course Overview

Course Title
Computer Programming

Subject Code
CS - Computer Science

Course Number
2313

Academic Unit
Computer Science (CS)

College/School
College of Engineering (EG)

Course Duration
One Semester

Credit Units
3

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

Medium of Instruction
English

Medium of Assessment
English

Prerequisites
CS1103B Media Computing or CS1303 Introduction to Internet and Programming or equivalent

Precursors
Nil

Equivalent Courses
Nil

Exclusive Courses
CS1315 Introduction to Computer Programming
CS2310 Computer Programming
CS2311 Computer Programming
CS2315 Computer Programming
CS2360 Java Programming

Part II Course Details

Abstract

This course aims to equip the students with in-depth concepts and techniques of programming using a high-level object-oriented programming language and to develop practical skills in producing quality programs.

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 advanced skills like using recursion and dynamic data structures.	60	x	x	x
4	Design and construct well-structured programs with good programming practices.	15	x	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 self-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.

Teaching and Learning Activities (TLAs)

	TLAs	Brief Description	CILO No.	Hours/week (if applicable)
1	Lecture	Various programming concepts and techniques will be introduced, explained and demonstrated with examples.	1, 2, 3, 4	3 hours/week

2	Lab	The laboratory sessions are designed to enable the students to 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 hours/week
3	Assignments	The assignments are more challenging tasks compared with laboratory exercises. The students need to analyze the requirements and design programming solutions by applying and combining various techniques learnt from lectures and laboratory exercises. They are also required to implement their solutions as practical computer programs, and to explain their ideas/ algorithms using suitable presentation methods (e.g. a report, flowchart, etc.).	2, 3, 4	
4	Quiz	The quiz provides the students opportunities to reflect on their understanding of various programming concepts and problem-solving techniques. They are required to predict and explain the behaviour of programs involving various programming concepts and techniques, and develop and implement programs or program segments to solve problems.	1, 2, 3, 4	

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks
1	Quiz	1, 2, 3, 4	20	
2	Assignments	2, 3, 4	20	

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

Lab

Criterion

ABILITY to UNDERSTAND and APPLY object-oriented programming concepts and techniques by designing, analyzing, testing, and debugging 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

Assessment Task

Quiz

Criterion

ABILITY to EXPLAIN object-oriented programming concepts and 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
Quiz

Criterion
APPLY learnt knowledge to solve a simple programming task

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
ABILITY to APPLY learnt knowledge to solve a task by designing, analyzing, testing, and debugging 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, multidimensional arrays, file I-O, recursion, pointers and dynamic data structures, object-based programming: data abstraction, classes, and the class library; programming style, program testing, exception handling.

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, recursion, exception handling.
- 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. Pointers. Dynamic data structures (eg. Dynamic array, linked list).
- Program development practice
  Professional programming styles. Program testing. Program documentation.

Reading List

Compulsory Readings

	Title
1	C.K. Poon & Matthew Chang (2007). Concepts and Techniques in C++ Programming. McGraw Hill.

Additional Readings

	Title
1	Walter Savitc (2010). Absolute C++. Addison-Wesley, 4th edition.