CS2312: PROBLEM SOLVING AND PROGRAMMING

Effective Term
Semester A 2022/23

Part I Course Overview

Course Title
Problem Solving and Programming

Subject Code
CS - Computer Science
Course Number
2312

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
Nil

Precursors
CS2310 Computer Programming or
CS2311 Computer Programming or equivalent

Equivalent Courses
Nil

Exclusive Courses
Nil

Part II Course Details

Abstract

This course aims to provide extensive practice in problem-solving using different programming paradigm, which includes the object-oriented programming, functional programming, and logic programming. Students will learn the fundamental concepts and distinctive features in these programming paradigms. They will develop skills to abstract data and entities from the problem domain, build models, design solutions using different paradigm principles and strategies, and implement solutions in these programs. Students will also explore tools and best practices in programming.

Course Intended Learning Outcomes (CILOs)

	CILOs	Weighting (if app.)	DEC-A1	DEC-A2	DEC-A3
1	Identify and describe fundamental programming paradigm concepts.	10	x	x	
2	Abstract data and entities from the problem domain, build models and design software solutions using different programming paradigm principles and strategies.	20		x	
3	Implement the respective design of different programming paradigms in programs using a modern programming language to solve problems.	50		x	
4	Apply tools and best practices in different programming paradigms.	10	x	x	
5	Evaluate and critique program coding and design based on different programming principles.	10	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.

Teaching and Learning Activities (TLAs)

	TLAs	Brief Description	CILO No.	Hours/week (if applicable)
1	Lecture	All CILOs will be introduced, explained, and demonstrated through lectures.	1, 2, 3, 4, 5	

2	Tutorial	Students will practice with solving problems using pre-designed programs, helping them to gear up their ability and skills in all CILOs.	1, 2, 3, 4, 5	
3	Quiz	The quiz will check students' achievement of the learning outcomes. This will provide timely feedback on their learning progress.	1, 2, 3, 4, 5	
4	Assignments	The assignments will require students to solve challenging problems by designing and writing object-oriented programs. Assignments will serve as a learning and assessment tool.	1, 2, 3, 4, 5	

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks (e.g. Parameter for GenAI use)
1	Quiz	1, 2, 3, 4, 5	20	
2	Assignments	1, 2, 3, 4, 5	30	Some portion may be allocated to weekly exercises

Continuous Assessment (%)

50

Examination (%)

50

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

Assignments

Criterion

1.1 ABILITY to articulate a convincing rationale for strategies used to design a solution for problem solving for different programming paradigms.

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
Assignments

Criterion
1.2 ABILITY to construct a program which conform to the program design and specification.

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
Assignments

Criterion
1.3 ABILITY to discover, explore and apply tools and best practices in different programming paradigms.

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
Assignments

Criterion
1.4 ABILITY to evaluate programs with a critical mind based on different programming paradigm principles.

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
2.1 ABILITY to identify and explain the concepts of different programming paradigms.

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
2.2 ABILITY to articulate a convincing rationale for strategies used to design a solution for problem solving with different programming paradigms.

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
2.3 ABILITY to construct a program which conform to the program design and specification.

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
2.4 ABILITY to discover, explore and apply tools and best practices in different programming paradigms.

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
2.5 ABILITY to evaluate programs with a critical mind based on different programming paradigm principles.

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
3.1 ABILITY to identify and explain the concepts of different programming paradigms.

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
3.2 ABILITY to articulate a convincing rationale for strategies used to design a solution for problem solving with different programming paradigms.

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
3.3 ABILITY to construct a program which conform to the program design and specification.

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
3.4 ABILITY to discover, explore and apply tools and best practices in different programming paradigms.

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
3.5 ABILITY to evaluate programs with a critical mind based on different programming paradigm principles.

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
Problem solving in the object-oriented, functional programming, and logic programming paradigms.
Syllabus

- Problem solving and programming paradigms
  Nature of problem solving. Structured programming. Object-based programming. Object-oriented programming, functional programming, logic programming.
- Features of different programming paradigms
  Abstraction. Class. Encapsulation. Inheritance. Polymorphism. Functions. Relations. Datatypes. Recursive datatypes. Lambda calculus. Rules. Unification.
- Constructing programs
  Association. Generalization. Specialization. Delegation. Realization. Aggregation. Dynamic Binding and Static Binding. Lazy evaluation. Recursion. Tail-recursion. Data abstraction.
- Overview of programming languages
  Declarative, imperative and hybrid programming. General versus domain-specific languages. Translation from source to executable code: compilation, interpretation, intermediate code generation. Design and choice of programming paradigms and languages for problem solving.

Reading List

Compulsory Readings

	Title
1	Nil

Additional Readings

	Title
1	Object-Oriented Programming (OOP) in Python 3https://realpython.com/python3-object-oriented-programming/
2	David Mertz "Functional Programming in Python." O’ 'Relly 2016
3	David Barnes Object-Oriented Programming with Java: An IntroductionPrentice Hall, 2000
4	Richard L. Halterman Learning to Program with Python 2011
5	Bruce Frederiksen Applying Expert System Technology to Code Reuse with Pyke 2008