CS1302: INTRODUCTION TO COMPUTER PROGRAMMING

Effective Term
Semester A 2024/25

Part I Course Overview

Course Title
Introduction to Computer Programming

Subject Code
CS - Computer Science
Course Number
1302

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
Nil

Exclusive Courses
CS1102 Introduction to Computer Studies

Part II Course Details

Abstract

This course aims to introduce to students key concepts, techniques, and good practices of programming using Python.

Course Intended Learning Outcomes (CILOs)

	CILOs	Weighting (if app.)	DEC-A1	DEC-A2	DEC-A3
1	Explain the structure of a computer program.	10	x	x	
2	Interpret, test and debug computer programs.	20	x	x	
3	Apply proper programming techniques to solve a task.	50		x	
4	Construct well-structured programs.	20		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 explore various programming concepts and techniques, explain and demonstrate the techniques with concrete examples.	1, 2, 3, 4	3 hours per week
2	Lab	Students will learn to apply programming to solve problems in practical scenarios. They will discover initial solution ideas to the problems and try testing some programs using an interative AI-assisted literate programming environment.	1, 2, 3, 4	1 hour per week

3	Assignment	Students will put theory into practice and get proficient in AI-assisted programming. They will design solutions to solve practical programming problems based on techniques and solution ideas from the lectures and labs.	2, 3, 4	

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks (e.g. Parameter for GenAI use)
1	Quiz	1, 3, 4	20	Correctly explain different parts of a computer program and the behaviour of its execution. Find out program errors and make corrections. Apply proper programming techniques to solve a task. Construct well-structured programs.
2	Assignment	2, 3, 4	30	Individual or group assignments on program development and testing.

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

Quiz

Criterion

1.1 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
2.1 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
3.1 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

The development of algorithms. Program design. Programming language. Control structures. Data types. Arrays/matrices, functions and parameters. Composite data types. List comprehension. Mathematics libraries. Structured decomposition. Programming style. Program testing. Introduction to recursion. Fundamentals on computer hardware architecture. Applications of programming to mathematical problem-solving methods.

Syllabus

- Program development environment
  Software development process. Development tools. Program development environments.
- Programming techniques and the development of algorithms
  Modular decomposition and stepwise refinement, principles of abstraction. Algorithms, the realisation of algorithms as programs. Program design: programming language, procedural abstraction, parameter-passing, control structures, iteration, recursion.
- Data structures
  The concept of data types. Simple data types. Arrays/matrices. List comprehension. Strings. Composite data types.
- Program development practice
  Elements of programming style. Program testing. Program documentation.

Reading List

Compulsory Readings

	Title
1	Richard L. Halterman (2018). Fundamentals of Python Programming
2	Allen Downey (2015). Think Python – How to Think Like a Computer Scientist. 2nd ed
3	Eric Matthes (2015). Python Crash Course: A Hands-On, Project-Based Introduction to Programming.
4	Paul Barry (2016). Head First Python: A Brain-Friendly Guide 2nd ed

Additional Readings

	Title
1	LinkedIn learning courses on AI-assisted literate programming tools such as Github Copilot and Jupyter. Available online at https://www.cityu.edu.hk/its/services-facilities/online-courses-linkedin-learning
2	John Denero (2011). Structure and Interpretation of Computer Programs. Available online at: https://wizardforcel.gitbooks.io/sicp-in-python
3	A Byte Of Python, by C.H. Swaroop. Available online at: https://python.swaroopch.com/
4	Guttag, John V (2021). Introduction to computation and programming using Python: with application to computational modeling and understanding data.