**King Saud University**
**College of Computer and Information Sciences**
**Department of Computer Science**

# CSC 111 Syllabus

| | |
|---|---|
| **Academic Year:** | First Semester - 1448 H |
| **Course:** | CSC111- Computer Programming - I |
| **Credit hours:** | 4 |
| **Prerequisites:** | CT101 |
| **co- requisite:** | None |

**Course Description:**

This course introduces fundamental programming concepts and systematic problem-solving techniques. It emphasizes algorithmic thinking using algorithms, pseudocode, and flowcharts before transitioning to structured programming in Python. Students learn to analyze problems, design solutions, and implement.

**Course Objectives:**

By the end of the course, students will be able to:

- Design step-by-step solutions using algorithms, pseudocode, and flowcharts before translating them into executable programs.
- Implement algorithmic solutions in Python, using appropriate control structures, data types, and built-in operators.
- Develop readable, well-structured, and documented programs, following Python naming conventions and best coding practices.
- Employ functions to achieve modularity and code reuse, including the use of parameters, return values, default arguments, and variable-length arguments.
- Apply Python's core data structures (lists, tuples, sets, dictionaries, and arrays) to organize and manipulate data effectively.
- Use abstraction mechanisms, such as functions and simple classes, to manage program complexity.

**Course Outcomes:**

Upon completing this course, students should have the following capabilities:

1. Analyze simple computing problems and design algorithms to solve them (K.1).
2. Apply the fundamentals of programming such as variables, conditional and iterative execution, functions, and data structures for implementing basic algorithms (K.2).
3. Understand basic concepts of object-oriented programming, such as defining classes and objects, setters and getters, etc. for solving simple programming problems (K.2).
4. Use an IDE for developing, debugging, testing and documenting programs (S.1).
5. Work on programming assignments that combine the different concepts and techniques learned in the course (S.2).

**Course Content:**

- Introduction to Programming and Problem Solving
- Flowcharts and Algorithm Design
- Decision and Control
- Loop Structures
- Introduction to Computer Programs
- Variables and Data Types
- Flowcharts to Python: Translation of Control Structures
- Functions
- Data Structures
- Recursion
- Introduction to OOP

**Assessment Methods and Grade Distribution:**

| # | Assessment task | Due Date | Total Assessment Score (%) |
|---|---|---|---|
| **1** | Midterm Exam 1 | Week 8 | 15 |
| **2** | Midterm Exam 2 | Week 11 | 15 |
| **3** | Lab Final Exam | Week 13 | 15 |
| **4** | Final Exam | 21 December 2026 (8:00-11:00) | 40 |
| **5** | Lab Evaluation | Weekly | 5 |
| **6** | Homework | Week7 & Week10 | 5 |
| **7** | Project | TBA | 5 |

**Course Policies:**

**Attendance**

Attendance in lectures, tutorials, and labs is mandatory. Students whose absence exceeds 25% may be denied entry to the final exam. Absence excuses must be submitted within one week.

**Submission and Deadlines**

All coursework must be prepared using appropriate software and submitted through the required channel by the stated deadline. Handwritten and late submissions are not accepted unless explicitly approved by the instructor.

**Plagiarism and AI Use**

Plagiarism, unauthorized use of AI tools, or submitting work that is not the student's own is strictly prohibited and will incur severe penalties. This includes the use of ChatGPT, Copilot, Gemini, Claude, or similar tools for assignments, projects, quizzes, or examinations.