**University of Management and Technology**

**School of Science and Technology**

**Course Outline**

**Semester: Fall 2023**

**Date: October 23, 2023**

|  |
| --- |
| **Code:** CC1021L |
| **Credit Hrs:** 1 |
| **Course Title:** Programming Fundamentals |
| **Prerequisites:** Matric level mathematics |
| **Grading Criteria:**  Quizzes: 10%  Graded Labs 10%  Assignments 10%  Midterm: 20 %  Project: 10%  Final: 40% |

**To Be Noted:** UMT has a wide mix of students including a majority with weak academic backgrounds in mathematics especially. This demands that the course is kept simple and core concepts are given more time to be properly absorbed. While the concepts are introduced slowly, practically solving a good number of simple to medium complexity problems will help in concept grasping. Therefore, this course should be less demanding mentally yet highly demanding practically. With this rationale in mind, the following course outline is suggested.

|  |  |  |
| --- | --- | --- |
| **Course Learning Outcomes (CLOs)** | | |
|  | **Domain** | **Bloom’s Taxonomy Level** |
| 1. Understand basic constructs of C++ | Cognitive | 2 |
| 1. Analyze a small-sized problem given as a requirement and design a logical solution | Cognitive | 4 |
| 1. Apply C++ constructs to develop a working program that implements the solution (in 2) | Cognitive | 3 |

**Course Outline**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course Coordinator** |  | **Contact** |  | |
| **Course**  **Description** | This course thoroughly introduces students with **problem-solving** with core programming concepts using the C++ programming language. | | | |
| **Course Learning**  **Outcomes** | At the end of the course, students should be able to comprehend a decent complexity problem, design a precise algorithmic solution using standard programming constructs, and convert it to working C++ code. In practical terms, the weakest passing student should be able to devise logic and working code, for a problem that requires arrays, loops and decisions. | | | |
| **Text**  **Book** | C++ How to Program, Dietel & Dietel | | | |
| **Reference Book:** | Starting out with C++, Tony Gaddis | | | |
| **Assignments** | 7 to 10 | **Quizzes** | | 5+ |
| **Mid Term**  **Examination** | 1 | **Final**  **Examination** | | 1 |
| **Project** | 1 |  | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tentative Plan** | | | |
| **Week** | **Course** | **Lab** | **Assessments** |
| Week 1 | Introduction to Computing Environment | Introduction to   1. IDE (Turbo C++, Dev C++/Code Blocks) 2. LMS 3. Online Portal |  |
| Week 2 | Problem-Solving Skills   1. Problem statement 2. Problem analysis chart to identify inputs, processing, and outputs) 3. Pseudo Codes 4. Flow Charts | 1. 1st Program “Hello World” 2. Project Create, Save 3. Submission on LMS | Assignment 1   1. Problem Solving Examples |
| Week 3 | Into to C++ Environment  Input / Output  Memory Variable  Data Types  Dry Run | 2 Sequential Problems in Lab   1. Problem understanding 2. Problem Solution 3. Solution to Code C++ 4. Common Errors 5. Debugging watches only 6. Filing | Assignment 2   1. Basic Elements 2. Identifiers 3. Pseudo Codes   **Quiz 1** |
| Week 4 | Type Casting  Relational Operators  Conditional Problems 1 | 3 Conditional Problems in Lab   1. Problem understanding 2. Problem Solution 3. Solution to Code C++ 4. Common Errors 5. Debugging watches only 6. Filing | Assignment 3   1. Relational Operators 2. Conditional 1 |
| Week 5 | Conditional Problems 2 | 3 Conditional Problems in Lab   1. Solution to Code C++ 2. Common Errors 3. Debugging watches/ Stepping 4. I/O on Files | Assignment 4   1. Conditional 2   **Quiz 2** |
| Week 6 | Iterative Problems | 3 Iterative Problems in Lab   1. Solution to Code C++ 2. Common Errors 3. Debugging 4. I/O on Files | Assignment 5   1. Iterative 1 |
| Week 7 | Nested Iterative Problems | 4 Nested Iterative Problems in Lab   1. Solution to Code C++ 2. Common Errors 3. Debugging 4. I/O on Files | Assignment 6   1. Iterative 2 2. Nested Iterative   **Quiz 3** |
| Week 8 | Modular Programming  Pass by Value  Pass by Reference | 1. Complex Mixed Problems using functions | Assignment 7   1. Problems Solving using Functions |
| Week 9 | 1. Arrays 2. Programming exercises | 1. Complex Mixed Problems | Assignment 8   1. Arrays 1 |
| **MID TERM – WEEK** | | | |
| Week 11 | 1. Arrays searching/sorting   Double dimensional arrays | 1. Programming exercises using complex arrays |  |
| Week 12 | Problem Solving 9   1. Pointers arithmetic | 1. Pointers Problems Complex | Assignment 9   1. Pointers 2. Dynamic Arrays |
| Week 13 | Problem Solving 10   1. Arrays vs Pointers 2. Dynamic Arrays 3. Strings | 1. Pointers vs Arrays 2. String Problems | Assignment 10   1. Pointers/ arrays 2. String Problems 3. STL functions of Strings   **Quiz 4** |
| Week 14 | 1. Stack & heaps | 1. Exercises Stack & heaps | Assignment 11 |
| Week 15 | Advanced Problem Solving   1. Complex Problems    1. Project    2. Competitions | 1. Complex Problems | Assignment 12   1. Complex Problems   **Quiz 5** |
|  | Advanced Problem Solving 2 | 1. Complex Mixed Problems |  |

**Sample Problems**

1. **Sequential Problems**
   1. Talal’s basic salary is input through the keyboard. His dearness allowance is 40% of basic salary, and house rent allowance is 20% of basic salary, calculate his gross salary.
   2. Instructors often assign different weights to exam scores when determining a student’s class grade. Typically, the final exam is assigned a higher weight than the midterm exam, even though the maximum possible score on the exams may be the same. For example, the midterm might be worth 40 % of the course grade, and the final worth 60%. The formula for the weighted course average in this case is

*Weighted course average = 40% of (midterm scores / midterm maximum) + 60% of (final score / final maximum)*

1. **Conditional Problem**
   1. You need to determine whether a meeting room is in violation of fire law regulations regarding the maximum room capacity. You will read in the maximum room capacity and the number of people to attend the meeting. If the number of people is less than or equal to the maximum room capacity, then you will announce that it is legal to hold the meeting and tells how many additional people may legally attend. If the number of people exceeds the maximum room capacity, then announce that the meeting cannot be held as planned due to fire regulations and tell how many people must be excluded in order to meet the fire regulations

1. **Iterative Problem**
   1. You are choreographing a circus show with various animals. For one act, you are given two kangaroos on a number line ready to jump in the positive direction (i.e, toward positive infinity).
      1. The first kangaroo starts at location x1 and moves at a rate of v1 meters per jump.
      2. The second kangaroo starts at location x2 and moves at a rate of v2 meters per jump.

You have to figure out a way to get both kangaroos at the same location at the same time as part of the show. If it is possible, say YES, otherwise NO.

1. **Nested Iterative Problem**
   1. So here comes an ineteresting task. Whole class was getting bor of the daily logic building tasks which sounds "interesting", Sir zeeshan decided to give a break and take you back to the class where we were drawing shapes. In this program you need to draw the diamond shape more interestingly.

The user will tell you the size of and you are to draw the diamond of that size. for example the diamond of size 5 is as follow



1. **Array Problem**
   1. A video player plays a game in which the character competes in a hurdle race. Hurdles are of varying heights, and the characters have a maximum height they can jump. There is a magic potion they can take that will increase their maximum jump height by 1 unit for each dose. How many doses of the potion must the character take to be able to jump all of the hurdles? If the character can already clear all of the hurdles output should be 0.

**Example**  
height = [1,2,3,3,2]

k=1

The character can jump 1 unit high initially and must take 3-1=2 doses of potion to be able to jump all of the hurdles.

1. **Complex Problems**
   1. Marie invented a Time Machine and wants to test it by time-traveling to visit Russia on the Day of the Programmer (the 256th day of the year) during a year in the inclusive range from 1700 to 2700.

From 1700 to 1917, Russia's official calendar was the Julian calendar; since 1919 they used the Gregorian calendar system. The transition from the Julian to Gregorian calendar system occurred in 1918, when the next day after January 31st was February 14th. This means that in 1918, February 14th was the 32nd day of the year in Russia.

In both calendar systems, February is the only month with a variable amount of days; it has 29 days during a leap year, and 28 days during all other years. In the Julian calendar, leap years are divisible by 4; in the Gregorian calendar, leap years are either of the following:

* + 1. Divisible by 400.
    2. Divisible by 4 and not divisible by 100.

Given a year, y, find the date of the 256th day of that year according to the official Russian calendar during that year. Then print it in the format dd.mm.yyyy, where dd is the two-digit day, mm is the two-digit month, and yyyy is y.

**For example,** the given year = 1984. 1984 is divisible by 4, so it is a leap year. The 256th day of a leap year after 1918 is September 12, so the answer is 12.09.1984.