**COMSATS University Islamabad,**

**Lahore Campus**

**□**

**Mid Term**

**□**

**Terminal Examination**

**–**

**Spring**

**2025**

**25**



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course Title: | Programming Fundamentals | | | | Course Code: | | CSC103 | Credit Hours: | | 4(3,1) |
| Course Instructor/s: | Dr. Rao Muhammad Adeel Nawab, Mr. Abdul Karim Shshid, Mr. Shahrukh Naeem, Mr. Akhzar Nazir | | | | Program Name: | | BS(CS), BS(SE) | | | |
| Semester: | 1st | Batch: | SP25-BCS/BSE | Section: |  | | Date: | | 14-04-2025 | |
| **Time Allowed:** | **1 Hour 30 Min** | | | | **Maximum Marks:** | | | | **40** | |
| Student’s Name: |  | | | | Reg. No. |  | | | | |
| **Important Instructions:**   * The exam consists of 3 questions, each with a different weightage. * All questions are compulsory and must be attempted. * You must solve all on the question paper (Do not forget to write your name and registration number on both) * Use answer book for dry run programs. * The exam is timed, and you will have 1 hour 30 min to complete all the questions. * Ensure that you allocate your time wisely across all the questions to maximize your score. * Write your answers in clear and concise language, using proper syntax and formatting where applicable (in C++ only). | | | | | | | | | | |

CLO-1 <Understanding> Demonstrate the fundamental concepts of programming.

Question 1:

(Part A): [1\*10=10 Marks]

Consider the following Real-world Problems and write down the most suitable Selection Structures (if, if-else, if-else-if) and Repetition Structures (for, while do-while, nested loop) for each Real-world Problem?

|  |  |  |
| --- | --- | --- |
| Sr. No. | Real Worl Problems | Most Suitable Structure |
| 1 | ATM Withdrawal Process  Input: Card Pin, Amount  Output: Success / Insufficient Balance / Incorrect PIN | if-else-if |
| 2 | Login Attempts Checker  Input: Username, Password  Output: Login success/failure (max 3 attempts) | while loop |
| 3 | Multiplication Table Generator  Input: Number (e.g., 5)  Output: 5x1 = 5 to 5x10 = 50 | for loop |
| 4 | Menu-driven Calculator  Input: Operation Choice  (1: Add, 2: Subtract, 3: Divide,)  Output: Result | if-else-if |
| 5 | Guess the Number Game  Input: User guesses  Output: Correct / Try again | do-while |
| 6 | Print Matrix:  Input: Rows, Columns  Output: Display Matrix | nested loop |
| 7 | Traffic Signal Decision System  Input: Signal Colour  Output: Stop / Wait / Go | if-else-if |
| 8 | Leap Year Checker  Input: Year  Output: Leap / Not Leap | if-else |
| 9 | Day Name from Number  Input: 1 to 7  Output: Monday to Sunday | if-else-if |
| 10 | Student Pass/Fail Decision  Input: Marks  Output: Pass / Fail | if-else |

(Part B): [1\*10=10 Marks]

Please Match Statements Column A with Options in Column B.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Column A | Opt. No | Column B | Correct Option No. From Column B |
| A | #include<iostream> | 1 | Used to take input from the user | **2** |
| B | cin >> | 2 | Directive to include standard input/output stream | **1** |
| C | if-else | 3 | Used to select one block from multiple choices | **8** |
| D | switch | 4 | The body that contains the code to be executed | **3** |
| E | for loop | 5 | Values passed into a function during definition | **7** |
| F | while loop | 6 | A placeholder to define function signature before main() | **9** |
| G | do-while loop | 7 | Best when the number of iterations is known | **10** |
| H | Function Prototype | 8 | Executes one block if condition true, another if false | **6** |
| I | Function Definition | 9 | Condition checked before execution of the loop body | **4** |
| J | Parameters | 10 | Loop that executes at least once | **5** |

# CLO2 <Applying>Employ programming constructs using a programming language

# Question.No.2) [2\*10=20 Marks]

Read the following program segments and write output for each segment (if any). If there are any error(s) in the program segment correct them and write the output.

|  |  |  |
| --- | --- | --- |
|  | Statement | Output |
| A. | int x = 0;  while (x++ < 3) {  cout << x << " ";  } | 1 2 3 |
| B. | int i, j;  for (i = 4; i > 0; i--) {  for (j = 0; j < i; j++) {  cout << "\*";  }  cout << endl;  } | \*\*\*\*  \*\*\*  \*\*  \* |
| C. | int x = 0, y = 1;  if (x = y) {  cout << "True" << endl;  } else {  cout << "False" << endl;  } | True |
| D. | int fun(int &x) {  x = x \* 2;  return x;  }  int main() {  int a = 4;  cout << fun(a) << " " << a << endl;  } | 8 8 |
| E. | int foo(int x) {  if (x == 0) return 0;  if (x > 0) return x;  return -x;  }  int main() {  int a = 4, b = -3, c = 0;  cout << foo(a) << " " << foo(b) << " " << foo(c) << endl;  } | 4 3 0 |
| F. | void oper(int num) {  int result = 0;  while (num != 0) {  result = result \* 10 + num % 10;  num /= 10;  }  cout << result << endl;  }  int main() {  oper(12345);  return 0;  } | 54321 |
| G. | void myFun(int &x, int &y) {  int temp = x;  x = y;  y = temp;  }  int main() {  int a = 5, b = 10;  myFun(a, b);  cout << a << " " << b;  return 0;  } | 10 5 |
| H. | int n = 2;  do {  cout << n \* n << endl;  --n;  } while (n >= 0);  cout << n << endl; | 4  1  0  -1 |
| I. | int main() {  for (int i = 1; i <= 4; i++) {  if (i == 2) {  continue;  }  cout << i << " ";  } | 1 3 4 |
| J. | int n;  cout << (n = 4) << endl;  cout << (n == 4) << endl;  cout << (n < 4) << endl;  cout << (n == 0) << endl;  cout << (n && 4) << endl;  cout << (!n) << endl; | 4  1  0  0  1  0 |