Introduction to Programming

Assignment 01

Due Date

Softcopy submission:- 03-Nov-2022; 23:59 hrs

Hardcopy submission:- 04-Nov-2022; 13:00 hrs

Instructions:

1. Write the assignment on A4 sheets.
2. Write both questions and answers in the assignment. It should be handwritten.
3. Clearly mention your Name, Registration Number, Semester, School and email ID at the beginning of the assignment sheet.
4. After you have completed the assignment, scan the assignment sheets as a single PDF, preferably using Adobe Scanner in mobile, and submit the softcopy of the assignment as PDF in the LMS within the Softcopy Submission deadline.
5. Submit the physical paper copy of the assignment on Hardcopy Submission deadline with multiple papers properly stapled.
6. The assignments will be graded only for those who have submitted both softcopy and hardcopy within the due date.
7. **There will not be any extension to the submission deadline**.

Note:

- All the assignment questions are from the reference textbook "**C in 21 Days** (**6th edition**)" which was shared to all students.
- Total mark for Assignment 01 is **46**. Incomplete assignments will be penalized.

Day-1 (**Pages 21-22**) [**10 Questions**]

1. Give three reasons why C is the best choice of a programming language.
2. What does the compiler do?
3. What are the steps in the Program Development Cycle?
4. What command do you need to enter in order to compile a program called **program1.c** with your compiler?
5. Does your compiler do both the linking and compiling with just one command, or do you have to enter separate commands?

6. What extension should you use for your C source files?

7. Is **FILENAME.TXT** a valid name for a C source file?

8. If you execute a program that you have compiled and it doesn't work as you expected, what should you do?

9. What is machine language?

10. What does the linker do?

Day-2 (**Page 39**) [**10 Questions**]

1. What is the term for a group of one or more C statements enclosed in braces?

2. What is the one component that must be present in every C program?

3. How do you add program comments, and why are they used?

4. What is a function?

5. C offers two types of functions. What are they, and how are they different?

6. What is the #include directive used for?

7. Can comments be nested?

8. Can comments be longer than one line?

9. What is another name for an include file?

10. What is an include file?

Day-3 (**Page-57**) (**Question numbers 4, 5 and 8 are discarded but preserve the question numbers as shown below**) [**6 Questions**]

1. What's the difference between an integer variable and a floating-point variable?

2. Give two reasons for using a double-precision floating-point variable (type **double**) instead of a single-precision floating-point variable (type **float**).

3. What are five rules that you know are always true when allocating size for variables?

6. What characters are allowed in C variable names?

7. What guidelines should you follow in creating names for variables and constants?

9. What's the minimum value that a type int variable can hold?

Day-4 (**Page-90**) [**10 Questions**]

1. What is the following C statement called, and what is its meaning?

```
x = 5 + 8;
```

2. What is an expression?

3. In an expression that contains multiple operators, what determines the order in which operations are performed?

4. If the variable x has the value 10, what are the values of x and a after each of the following statements is executed separately?

    a = x++;

    a = ++x;

5. To what value does the expression 10 % 3 evaluate?
6. To what value does the expression 5 + 3 * 8 / 2 + 2 evaluate?
7. Rewrite the expression in question 6, adding parentheses so that it evaluates to 16.
8. If an expression evaluates to false, what value does the expression have?
9. In the following list, which has higher precedence?

    a. == or <

    b. * or +

    c. != or ==

    d. >= or >

10. What are the compound assignment operators, and how are they useful?

Day 5 (**Page-121**) [**10 Questions**]

1. Will you use structured programming when writing your C programs?
2. How does structured programming work?
3. How do C functions fit into structured programming?
4. What must be the first line of a function definition, and what information does it contain?
5. How many values can a function return?
6. If a function doesn't return a value, what type should it be declared?
7. What's the difference between a function definition and a function prototype?
8. What is a local variable?
9. How are local variables special?
10. Where should the main( ) function be placed?
