# TUTORIAL 2

1)

These are created by placing two forward slashes (//) before the comment text. Anything written after // on the same line is considered a comment.

The purpose of comments in a program is to provide additional information and explanations to make the code more understandable to other programmers or to oneself in the future. Comments are not executed by the computer and do not affect the functionality of the program. They are used to document the code, describe the purpose of specific sections or variables, and improve code readability and maintainability. 2)

The essential function in a C program is the main() function. 3)

The purpose of the scanf() function is to read input from the user.

4)

Yes

5)

(a) record1

(f) name

(i) name\_and\_address

1. 1record: Identifiers cannot start with a digit.
2. file-3: Identifiers cannot contain hyphens.

(e) $tax: Identifiers cannot start with a dollar sign.

(g) name and address: Identifiers cannot contain spaces.

(j) 123 - 45 - 6789: Identifiers cannot contain spaces or hyphens.

6)

1. False. By default, printf does not automatically start printing at the beginning of a new line unless specified.
2. False. Comments are ignored by the compiler and do not produce any output during program

execution.

1. True. The escape sequence \n moves the cursor to the beginning of the next line.
2. True. All variables must be declared and defined before they are used in the program.
3. True. In C, variables must have a type specified when they are defined.
4. False. C is a case-sensitive language, so number and NuMbEr are considered different variables.
5. False. The number of printf statements does not determine the number of lines of output. Multiple lines can be printed using a single printf statement with appropriate formatting.

7)

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

8)

1. scanf( "%d", &value );
2. printf( "The product of %d and %d is %d\n", x, y, product );
3. scanf( "%d", &anInteger );
4. printf( "Remainder of %d divided by %d is %d\n", x, y, x % y );
5. printf( "The sum is %d\n", x + y );
6. printf( "The value you entered is: %d\n", value );

9)

1. 2
2. 4
3. x=
4. x=2
5. 5 = 5

f Nothing

1. Nothing.
2. Nothing
3. Nothing

10)

1. True. C operators are evaluated from left to right.
2. True. All the given variable names are valid.
3. False. The statement printf("a = 5;"); is an example of a print statement, not an assignment statement.
4. True. In a valid arithmetic expression without parentheses, the evaluation is performed from left to right.
5. False. The variable names 3g, 87, 67h2, h22, and 2h are invalid because they start with a digit. Variable names must start with a letter or an underscore.