**NADUNI KODIKARA**

**29861**

**23.1**

# TUTORIAL 02

1. **How do you write comments in a c program? What is the purpose of comments in a program?**

* + Single comment - ex :- // comment
  + Multi-line comment – ex:- /\* comment\*/
  + We write comments in c programming using double slash commonly. It helps the programmer to add comments to the specific area regarding any required detail.

1. **Which is the function that is essential in a C program?**

•The main function is the essential function in C program.

#include<stdio.h>

1. **What is the purpose of ‘scanf’ ?**

• Scanf function is used to take input from the user.

1. **Is ‘standard c’ a case sensitive language?**

• Yes, it is a case sensitive language. In c language the case of each variable varies, therefore it helps to compile more faster.

1. **Determine which of the following are valid identifiers. If invalid, explain why.**

1. record1 - Valid
2. 1record - Invalid . Because A variable cannot start with a Number.
3. file-3 - Invalid. Because it can’t have special characters like “ –“ it is

incorrect

1. return - Valid
2. $tax - Invalid . Because a variable cannot have special characters. (f) name - Valid
3. name and address - Invalid. Because a variable name cannot have keywords and spaces.
4. name-and-address - Invalid . Because cant use special characters .
5. name\_and\_address – Valid
6. 123 - 45 - 6789 - Invalid . Because variable can’t start with numbers and also cant use Special characters.

**6. State whether each of the following is true or false. If false, explain why.**

1. Function printf always begins printing at the beginning of a new line.

* + **False** .

* + Printf function doesn’t print at the beginning of a new line its print when ever the cursor begins.If you need to get a new line you have to write the code with “\n” this is called as line breaker.

1. Comments cause the computer to print the text enclosed between /\* and \*/ on the screen when the program is executed.

* + **False**

* + These comments doesn’t show when a program is executed. Comments are used for programmers to identify the coding parts easily.

1. The escape sequence \n when used in a printf format control string causes the cursor

to position to the beginning of the next line on the screen.

* + **True**

1. All variables must be defined before they’re used.

* + **True**

1. All variables must be given a type when they’re defined.

* + **True**

1. C considers the variables, number and NuMbEr to be identical.

* + **False**

* + number and NuMbEr should be same in the C language because C language is a Case sensitive language . So The both should be in lower case or uppercase if not they will be incorrect.

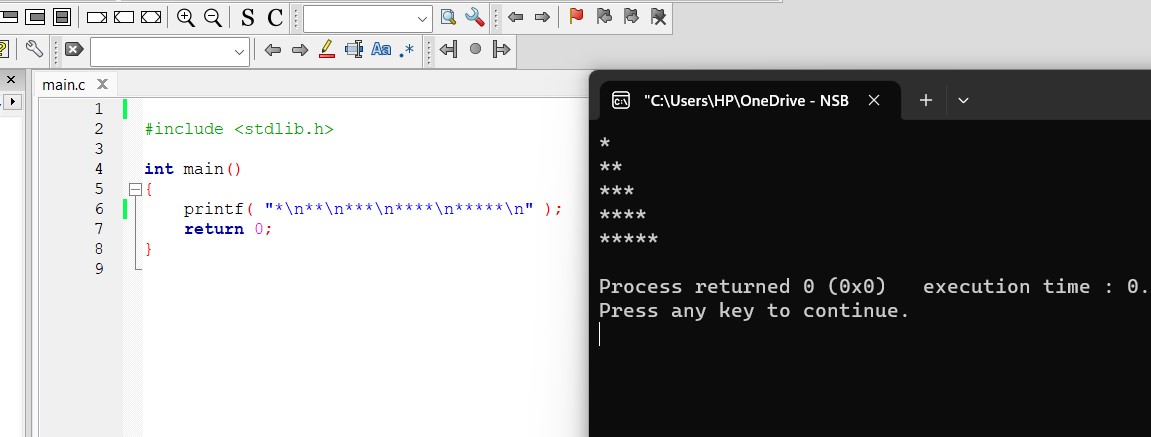
1. A program that prints three lines of output must contain three printf statements.

* + - **False**

* + - No need of three lines of printf statements to get three line of output in one printf statement can get three lines of outputs by using “\n” line breaker.

1. **What does the following code print?**

printf( "\*\n\*\*\n\*\*\*\n\*\*\*\*\n\*\*\*\*\*\n" );



1. **Identify and correct the errors in each of the following statements. (Note: There may be more than one error per statement.)**

1. scanf( "d", value );

* + scanf ( “%d”, &value);

1. printf( "The product of %d and %d is %d"\n, x, y );

* + printf( "The product of %d and %d is %d \n” x, y );

1. Scanf( "%d", anInteger );

* + Scanf( "%d", &anInteger );

1. printf( "Remainder of %d divided by %d is\n", x, y, x % y ); • printf( "Remainder of %d divided by %d is %d \n" x, y, x % y );’

1. print( "The sum is %d\n," x + y );

* + printf( "The sum is %d \n" x + y );

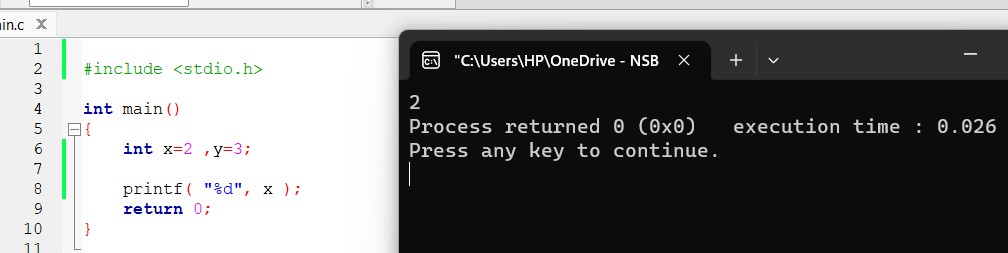
1. Printf( "The value you entered is: %d\n, &value );

* + printf( "The value you entered is %d \n” ,value );

**9. What, if anything, prints when each of the following statements is performed? If nothing \prints, then answer “Nothing.” Assume x = 2 and y = 3 .**

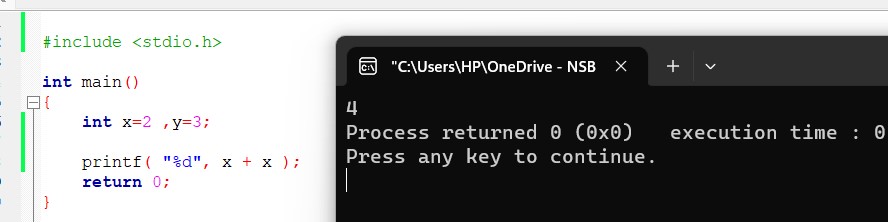
1. printf( "%d", x );

* + 2



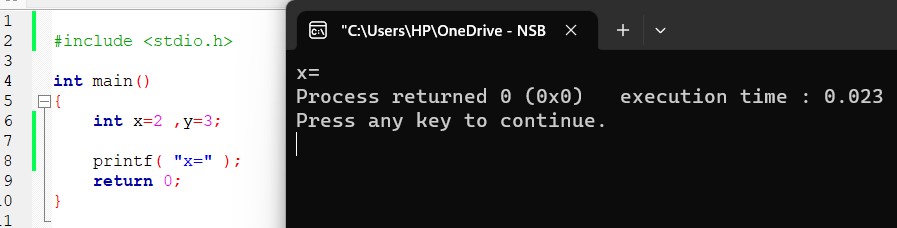
1. printf( "%d", x + x );

* + 4



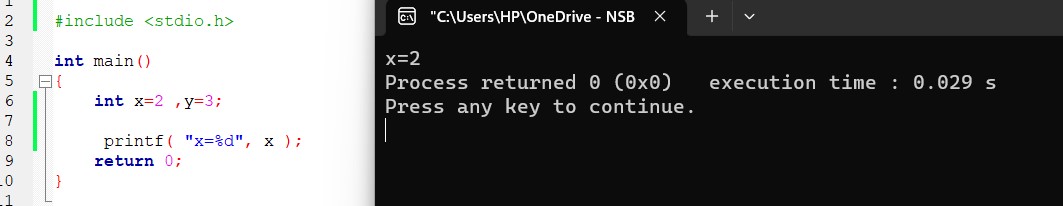
1. printf( "x=" );

* + x=



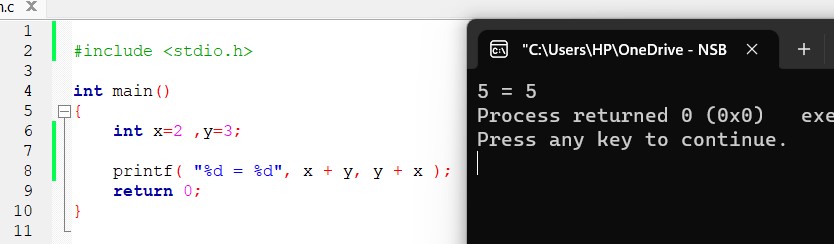
1. printf( "x=%d", x );

* + x= 2



1. printf( "%d = %d", x + y, y + x );

* + 5=5



1. z = x + y; = Nothing

1. scanf( "%d%d", &x, &y ); = Nothing

1. /\* printf( "x + y = %d", x + y ); \*/ = Nothing

1. printf( "\n" ); = Nothing

**10. State which of the following are true and which are false. If false, explain your answer.**

1. C operators are evaluated from left to right.

* + **False**
  + C operators are not always evaluated from left to right These are evaluated from precedence and associativity.

1. The following are all valid variable names: \_under\_bar\_ , m928134 , t5 , j7 , her\_sales

his\_account\_total , a , b , c , z , z2 .

* + **True**

1. The statement printf("a = 5;"); is a typical example of an assignment statement.

* + **False .**
  + It is not a assignment statement. If it is a assignment statement there must not be a printf statement.

1. A valid arithmetic expression containing no parentheses is evaluated from left to right.

* + **True**

f) The following are all invalid variable names: 3g , 87 , 67h2 , h22 , 2h

* **False**
* Only 3g,87,67h2,2h are invalid variable names .