Date:
Page No.:

INTRODUTION TO PROGRAMMING

Assignment - 1

Submitted to:

Renu Taneja

Submitted by:
Vishal Parashar
03551203116
IT (eve)
2nd Semester

Note: Answer in 3-4 lines (each question)

Q.1 What is Global variable?

In computer programming, a global variable is a variable with global scope, meaning that it is visible (hence accessible) throughout the program, unless shadowed.

Q.2 How does a user find coding errors?

Import <error.h> header file, user can find various error codes contained in this file. So, a programmer can check the returned values and can take appropriate action depending on the return value errno

Date:
Page No.:

Q.3 What are Bitwise operators?

In the C programming language, operations can be performed on a bit level using bitwise operators. A bitwise operation operates on one or more bit patterns or binary numerals at the level of their individual bits. It's a fast, simple action directly supported by the processor, and is used to manipulate values for comparisons & calculations.

Q.4 What is getchar()?

It reads a single character of input & returns that character as the value of function. If there is an error reading the character, or if the end of input is reached, getchar() returns a speval value ('EOF').

Q.5 What is syntax error?

A syntax error is an error in the syntax of a sequence of characters or token that is intended to be written in a particular programming language. Syntax errors are detected at compile time. eg:- >include <stdio.h> ✗
include<stdio.h> ✓

Q.6 What is function?

A function is a group of statements that together perform

a task. Every C program has at least one function which is main(), and all the most trivial programs can define additional functions. Code can be divide into separate functions

Eg :- main(), min(), etc.

Q.7 What are the different types of loops used in C?

A loop statement allows us to execute a statement or group of statements multiple times.

While loop :- Represents (Repeats) a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.

For loop :- Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.

Do-while loop :- It is more like a while statement, except that it tests the condition at the end of loop body.

Q.8 Explain # include

In C programming language, the # include directive tells the preprocessor to insert the contents of another file into the source code at the point where the # include directive is found. Include directives are typically used to include the C header files for C functions that are held outside of the current source file.

Syntax → # include <header file>

Q.9 What is the purpose of Keyword void?

When used as a function return type : the void keyword specifies that the function does not return a value.

→

Date :
Page No. :

when used for a function's parameter list : void specifies that the function takes no parameters.

Q.10 Write a program to find largest no. from two numbers?

```
#include <stdio.h>
#include <conio.h>
int main()
{  int num1, num2;
   printf ("Enter two numbers:");
   scanf ("%d %d", &num1, &num2);
   if (num1 > num2)
   {  printf ("%d is maximum", num1);
   }
   else
   {  printf ("%d is maximum", num2);
   }
   getch();
}
```

Q.11 Differentiate between = and == ? Give examples.

'=' is an assignment operator ; it is used to assign the value of variable or expression , while '==' is an equal to operator is a relational operator used for comparison (value comparison of both left and right side operands).

Equal to (=) → Assignment

```
x = a+b;
y = x;
```

Here, when first expression evaluates value of (a+b) will be assigned into x and in second expression, to y.

Equal to (==) →

```
int x, y;
x = 10;
y = 10;
if (x == y)
    printf ("True");
else
    printf ("False");
```

when expression x == y evaluates, it will return 1 (it means condition is true) and TRUE will print.

Q.12 Explain Continue and Break statement

Break statement → when this statement is encountered inside a loop, the loop is immediately terminated and the program control resume at the next statement following the loop. It can be used to terminate a case in the switch statement.

Continue statement → Continue statement is mostly used inside loops. Whenever it's encountered inside a loop, control directly jumps to the beginning of the loop for next iteration, skipping the execution of statements inside loop's body for the current iteration.

Q.13 Differentiate between calling function and called function.

Difference between call by value and call by reference. In call by value, a copy of actual arguments is passed to formal arguments of the called function and any change made to the formal arguments in called fn have no effect on values on the values of actual arguments in calling

Date:
Page No.:

Q.14 Define Array. How 2D array is initialized?

An array is a collection of data items, all of the same type, accessed using a common name.
A 2-D array is like a table. For initializing 2D array, we need to assign values to each elements of an array.
int a[3][2] = { {1,4}, {5,2}, };

Q.15 Define arithmetic operators. List 5 of them.

An arithmetic operator is a mathematical function that takes two operands & performs a calculation on them.
+ → Adds two operands → A+B = 30
- → Subtracts second operand from the first → A-B = 1
* → Multiplies both operands → A*B = 200
/ → Divides Numerator by denominator → A/B = 20
% → Modulus operator and remainder → B%A = 0

Q.16 List applications of C language.

(i) Used for creating computer applications
(ii) Used in writing embedded softwares
(iii) Used to implements different operating system operat
(iv) Firmware for various electronics, industrial and communications products which use micro-controllers.

Q.17 What is Static variable and what is it's Scope ?

In the C programming language, static is used with global variables and functions to set their scope to the containing file. In local variables, static is used to store the variable in the statically allowed memory instead of automatically allocated memory.