Unit - III

CHAPTER - 11

Functions in C++

Part – 1

Choose the best answer

1. Which of the following header file defines the standard I/O predefined functions ?
   A) stdio.h B) math.h C) string.h D) ctype.h
2. Which function is used to check whether a character is alphanumeric or not.
   A) isalpha() B) isdigit() C) isalnum() D) islower()
3. Which function begins the program execution ?
   A) isalpha() B) isdigit() **C) main()** D) islower()
4. Which of the following function is with a return value and without any argument ?
   A) x=display(int, int) **B) x=display()**
   C) y=display(float) D) display(int)
5. Which is return data type of the function prototype of add(int, int); ?
   A) int B) float C) char D) double
6. Which of the following is the scope operator ?
   A) > B) & C) % **D) ::**

Part –II

Answer to all the questions (2 Marks):

1. **Define Functions.**
   A large program can typically be split into smaller sized blocks called as functions. Where each sub-program can perform some specific functionality.

2. **Write about strlen() function.**
   The **strlen()** takes a null terminated byte string source as its argument and returns its length. The length does not include the null(\0) character.

Ex:

```
name= "vijay"
strlen(name);        [ length of the name = 5 ]
```

3. **What is importance of void data type.**

void type has two important purposes:

- To indicate the function does not return a value
- To declare a generic pointer.

4. **What is Parameter and list its types?**
   Arguments or parameters are the means to pass values from the calling function to the called function

.

1. The variables used in the function definition as parameters are known as **formal parameters.**
2. The constants, variables or expressions used in the function call are known as **actual parameters.**

5. Write a note on Local Scope.

- A local variable is defined within a block. A block of code begins and ends with curly braces { }.
- A local variable cannot be accessed from outside the block of its declaration.

Part – III

Answer to all the questions (3 Marks):

1. What is Built-in functions?

- The functions which are available by default known as **"Built-in"** functions
- Header files provide function prototype and definitions for library functions.
- header file can be identified by their file extension **.h.**
- A single header file may contain multiple built-in functions.

Example: **stdio.h** -- it contains pre-defined **"standard input/output"** functions.

2. What is the difference between isuppr() and toupper() functions ?

isuppr()	toupper()
This function is used to check the given character is uppercase.	This function is used to convert the given character into its uppercase.
This function will return 1 if true otherwise 0.	This function will return the upper case equivalent of the given character.
Ex: **int n=isupper('A');**	Ex: **char c = toupper('k'); -- K**

3. Write about strcmp() function.

The **strcmp()** function takes two arguments: string1 and string2. It compares the contents of string1 and string2 lexicographically.

The strcmp() function returns a:

- Positive value if the first differing character in string1 is greater than the corresponding character in string2.
- Negative value if the first differing character in string1 is less than the corresponding character in string2.
- 0 if string1 and string2 are equal.

4. Write short note on pow() function in C++.

The **pow()** function returns base raised to the power of exponent.

If any argument passed to **pow()** is long double, the return type is promoted to long double.

If not, the return type is double. The **pow()** function takes two arguments:

base - the base value
exponent - exponent of the base

5. What are the information the prototype provides to the compiler?

Example: ***long fact (int, double)***

The prototype above example provides the following information to the compiler:

- The return value of the function is of type long.
- **fact** is the name of the function.
- The function is called with two arguments:
  - The first argument is of int **data** type.
  - The second argument is of **double** data type.

6. What is default arguments? Give example.

In C++, one can assign default values to the formal parameters of a function prototype. The Default arguments allows to omit some arguments when calling the function.

When calling a function,

- For any missing arguments, complier uses the values in default arguments for the called function.
- The default value is given in the form of variable initialization.

Example :
```
defaultvalue(x,y);
defaultvalue(200,150);
defaultvalue(150);
```

Part –IV
Answer to all the questions (5 Marks):

1. Explain Call by value method with suitable example.

Call by value method copies the value of an actual parameter into the formal parameter of the function. In this case, changes made to formal parameter within the function will have no effect on the actual parameter.

Example Program:

```
#include<iostream>
using namespace std;
void display(int x)
{
int a=x*x;
cout<<"\n\n The Value inside display function (a * a):"<<a;
}
int main()
{
int a;
cout<<"\n\n Enter the Value for A :";
cin>>a;
display(a);
cout<<"\n\n The Value inside main function "<<a;
return(0);
}
```

Output :

```
Enter the Value for A : 5
The Value inside display function (a * a) : 25
The Value inside main function 5
```

2. What is Recursion? Write a program to find GCD using recursion.

A function that calls itself is known as recursive function. And, this technique is known as recursion.

Example Program: (**Find GCD using recursion**)

```
#include <iostream>
using namespace std;
int gcd(int n1, int n2)
{
if (n2 != 0)
return gcd(n2, n1 % n2);
else
return n1;
}
int main()
{
int num1, num2;
cout << "Enter two positive integers: ";
cin >> num1 >> num2;
cout << "Greatest Common Divisor (GCD) of :" << num1;
cout<< " & " << num2 << " is: " << gcd (num1, num2);
return 0;
}
```

Output:

```
Enter two positive integers: 350 100
Greatest Common Divisor (GCD) of : 350 & 100 is: 50
```


3. What are the different forms of function return? Explain with example.

Returning from the function is done by using the **return** statement.

The **return** statement stops execution and returns to the calling function. When a **return** statement is executed, the function is terminated immediately at that point.

Different forms of function return:

1. *The return statement*
2. *The Returning values*
3. *The Returning by reference*

1. The return statement

The **return** statement is used to return from a function. It is categorized as a jump statement because it terminates the execution of the function and transfer the control to the called statement.

A **return** may or may not have a value associated with it. If return has a value associated with it, that value becomes the return value for the calling statement.

Even for void function return statement without parameter can be used to terminate the function.

Syntax:

return expression / variable;

Example : return(a+b); return(a);
return;

2. The Returning values:

The functions that return no value is declared as void. The data type of a function is treated as **int**, if no data type is explicitly mentioned.

Example :

```
int add (int, int);
add (int, int);
```

In both prototypes, the return value is int, because by default the return value of a function in C++ is of type **int**. Look at the following examples:

3. The Returning by reference

```
#include<iostream>
using namespace std;
int main()
{
int n1=150;
int &n1ref=n1;
cout<<"\n The Value of N1 = "<<n1<<" and n1Reference = "<<n1ref;
n1ref++;
cout<<"\n After n1 increased the Value of N1 = "<<n1;
cout<<" and n1Reference = "<<n1ref;
return(0);
}
```


Output:

The Value of N1 = 150 and n1Reference = 150
After n1 increased the Value of N1 = 151 and n1Reference = 151

4. Explain scope of variable with example.

Scope refers to the accessibility of a variable. There are four types of scopes in C++.

1. ***Local scope***
2. ***Function scope***
3. ***File scope***
4. ***Class scope***

1. Local scope

Inside a block which is called local variables.

Example:

```
#include<iostream>s
using namespace std;
int main ( )
{
int a, b ;
a = 10;
b = 20;
if (a > b)
{
```

```
int temp;
temp = a;
a = b;
b = temp;
}
cout <<"\n Descending order .... \n";
cout <<a <<"\t"<<b;
return(0);
}
```

Output:

```
Descending order ….
10      20
```

2. **Function scope**

Inside a function is called function variables.

Example:

```
#include<iostream>
using namespace std;
void add(int x, int y)
{
int m=x+y; //'m' declared within function add()//
cout<<"\nThe Sum = "<<m;
}
int main ( )
{
int a, b ;
a = 10;
b = 20;
add(a,b);
return(0);
}
```


3. **File scope**

Outside of all functions which is called global variables.

Example:

```
#include<iostream>
using namespace std;
int file_var=20; //Declared within File//
void add(int x, int y)
{
int m=x+y+file_var;
cout<<"\n The Sum = "<<m;
}
int main ( )
{
int a, b ;
a = 10;
b = 20;
add(a,b);
cout<<"\nThe File Variable = "<<file_var;
return(0);
}
```

Output:

```
The Sum = 50
The File Variable =20
```

4. Class scope

Inside a class is called class variable or data members.

class student
{
private :
int mark1, mark2, total;
};
	The class student contains mark1, mark2 and total are data variables. Its scope is within the class student only.

5. Write a program to accept any integer number and reverse it.

```
#include <iostream>
using namespace std;
int main()
{
   int n, reversedNumber = 0, remainder;
   cout << "Enter an integer: ";
   cin >> n;
   while(n != 0)
   {
      remainder = n%10;
      reversedNumber = reversedNumber*10 + remainder;
      n /= 10;
   }
   cout << "Reversed Number : " << reversedNumber;
   return 0;
}
```


Output:

Enter an integer: 12345

Reversed number: 54321

M.VIJAYA KUMAR, MCA.,M.Phil.,B.Ed.,PGDCA.,
PGT-COMPUTER TEACHER,

V.SUJITHA, MCA.,M.Phil.,
COMPUTER TEACHER,

SASTRA MATRICULATION HIGHER SECONDARY SCHOOL,
KILPENNATHUR, TIRUVANNAMALAI 604601

Mobile No : 9655826843 ***Email : vijay28soft@gmail.com***

*****ALL THE BEST*****