Chapter 9

1. What is function? Write advantage of function.

A **function** is a block of organized, reusable code that is used to perform a single, related action. Here are several **advantages** of using **functions:**

- Use of **functions** enhances the readability of a program
- A big code is always difficult to read. Breaking the code in smaller **Functions** keeps the program organized
- Easy to understand and makes it reusable.
- A function can be used to create our own header file which can be used in any number of programs

2. What is modular programming? Write some characteristics of modular programming.

Modular Programming allows development to be divided by splitting down a **program** into smaller programs in order to execute a variety of tasks.

characteristics of modular programming:

- Development Can be Divided
- Readable Programs
- Programming Errors are Easy to Detect
- Allows Re-Use of Codes
- Improves Manageability
- It is helpful for industrial level software design and collaboration among developers

3. What is parameter (argument)?

Parameters allow us to pass information or instructions into functions and procedures. **Parameters** are the names of the information that we want to use in a function or procedure. The values passed in are called arguments.

4. Describe different categories of functions

5. What is recursion? What is static value?

Recursion is the process of repeating items in a self-similar way. In programming languages, if a program allows you to call a function inside the same function, then it is a **recursive** call of the function.

Static variables can be defined inside or outside the function. They are local to the block. The default value of **static variables** is zero. The **static variables** are alive till the execution of the program.

6. write a program that calculates the factorial of n number

```c
#include <stdio.h>
void main(){
  int i,f=1,num;

  printf("Input the number : ");
  scanf("%d",&num);

  for(i=1;i<=num;i++)
      f=f*i;

  printf("The Factorial of %d is: %d\n",num,f);
}
```

7. Write a program to calculate the factorial of a number using recursion

```c
#include<stdio.h>
long int multiplyNumbers(int n);
int main() {
    int n;
    printf("Enter a positive integer: ");
    scanf("%d",&n);
    printf("Factorial of %d = %ld", n, multiplyNumbers(n));
    return 0;
}

long int multiplyNumbers(int n) {
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}
```

8. Write a program to calculate Fibonacci series

```c
#include <stdio.h>
int main() {
    int t1 = 0, t2 = 1, nextTerm = 0, n;
    printf("Enter a positive number: ");
    scanf("%d", &n);

    // displays the first two terms which is always 0 and 1
    printf("Fibonacci Series: %d, %d, ", t1, t2);
    nextTerm = t1 + t2;

    while (nextTerm <= n) {
        printf("%d, ", nextTerm);
        t1 = t2;
        t2 = nextTerm;
        nextTerm = t1 + t2;
    }

    return 0;
}
```

9. Write a program to check whether a number is prime or not

```c
#include <stdio.h>
int main() {
    int n, i, flag = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &n);

    for (i = 2; i <= n / 2; ++i) {

        // condition for non-prime
        if (n % i == 0) {
            flag = 1;
            break;
        }
    }

    if (n == 1) {
        printf("1 is neither prime nor composite.");
    }
    else {
        if (flag == 0)
            printf("%d is a prime number.", n);
        else
            printf("%d is not a prime number.", n);
    }

    return 0;
}
```

Chapter 10

1. What is Structure?

A **structure** is a user defined data type in **C**/**C**++. A **structure** creates a data type that can be used to group items of possibly different types into a single type. In C programming, a struct (or structure) is a collection of variables (can be of different types) under a single name.

```
struct Person
{
    char name[50];
    int citNo;
    float salary;
};
```

Here, a derived type ‘struct Person’ is defined. Now, we can create variables of this type.

2. How to declare structure in c?

A structure variable can either be declared with structure declaration or as a separate declaration like basic types.

```
struct Person
{
    char name[50];
    int citNo;
    float salary;
};
```

Here, a derived type ‘struct Person’ is defined. Now, we can create variables of this type.

3. Differentiate between arrays and structures.

Structure can be defined as a data structure used as container which can hold variables of different types. On other hand Array is a type of data structure used as container which can hold variables of same type and do not support multiple data type variables.

Structure do not have concept of Pointer internally. On other hand in case of Array it internally implements Pointer which always points to the first element of Array.

Structure object can be created after declaration later in the program. On other hand in case of Array we can't create its object after declaration.

Structure supports multiple data-type variables as input. On other hand in case of Array we can't have different data-type variable as input as it only supports same type data variables.

4. What is union?

A union is a special data type available in C that allows to store different data types in the same memory location. You can define a union with many members, but only one member can contain a value at any given time. Unions provide an efficient way of using the same memory location for multiple-purpose.

5. Differentiate between Union and structure

Structure is the container defined in C to store data variables of different type and also supports for the user defined variables storage. On other hand Union is also similar kind of container in C which can also holds the different type of variables along with the user defined variables.

In Structure multiple members can be can be initializing at same time. On other hand in case of Union only the first member can get initialize at a time.

As mentioned in definition Structure do not have shared location for its members so size of Structure is equal or greater than the sum of size of all the data members. On other hand Union does not have separate location for each of its member so its size or equal to the size of largest member among all data members.

Syntax of declare a Structure in C is as follow:

```
struct struct_name{
   type element1;
   type element2;
   .
   .
} variable1, variable2, ...;
```

On other syntax of declare a Union in C is as follow:

```
union u_name{
   type element1;
   type element2;
   .
   .
} variable1, variable2, ...;
```

Chapter 11

1. What is pointer? Write down the benefit of using pointers.

A pointer is a variable whose value is the address of another variable. Here is some benefit of using pointers in program:

- It allows management of structures which are allocated memory dynamically
- It allows passing of arrays and strings to functions more efficiently.
- It makes possible to pass address of structure instead of entire structure to the functions
- It makes possible to return more than one value from the function.

2. Write down the declare and initialization of pointer.

Declaring a Pointer

Like variables, pointers in C programming have to be declared before they can be used in your program. Pointers can be named anything you want as long as they obey C's naming rules. A pointer declaration has the following form.

```
data_type * pointer_variable_name;
```

Here,

data_type is the pointer's base type of C's variable types and indicates the type of the variable that the pointer points to.

The asterisk (*: the same asterisk used for multiplication) which is indirection operator, declares a pointer.

Initialize a pointer

After declaring a pointer, we initialize it like standard variables with a variable address. If pointers in C programming are not uninitialized and used in the program, the results are unpredictable and potentially disastrous.

To get the address of a variable, we use the ampersand (&)operator, placed before the name of a variable whose address we need. Pointer initialization is done with the following syntax.

Pointer Syntax

```
pointer = &variable;
```

3. What is swapping? Write a program that swap two values by using Pointers.

In computer programming the act of swapping variables refers to mutually exchanging the values of the variables. Here is program to swap 2 values by sing pointers:

```c
#include <stdio.h>

int main()
{
   int x, y, *a, *b, temp;

   printf("Enter the value of x and y\n");
   scanf("%d%d", &x, &y);

   printf("Before Swapping\nx = %d\ny = %d\n", x, y);

   a = &x;
   b = &y;

   temp = *b;
   *b = *a;
   *a = temp;

   printf("After Swapping\nx = %d\ny = %d\n", x, y);

   return 0;
}
```

4. What do you mean by call by value and call by reference?

The call by value method of passing arguments to a function copies the actual value of an argument into the formal parameter of the function. In this case, changes made to the parameter inside the function have no effect on the argument.

The call by reference method of passing arguments to a function copies the address of an argument into the formal parameter. Inside the function, the address is used to access the actual argument used in the call. It means the changes made to the parameter affect the passed argument.

This Note is made by Sakib Mahmud (Sigmakib)

