Functions

Function:- A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. Functions are used to perform certain actions, and they are important for reusing code: Define the code once, and use it many times.

Syntax:-

```
Function_type function_name()      //function declaration
{
Statement 1;
Statement 2;
…….
Statement 3;
}
```

Example:-

```
#include<iostream>
using namespace std;
void myfun()              //  function declaration
{
  cout << "I just got executed.";          //  function definition
}
int main()
{
  myfun();              //  function calling
  return 0;
}
```

Output:-

```
I just got executed.
```

Basically function four types.

1. No argument and no return type.
2. No argument but return type.
3. Argument but no return type.
4. Argument but return type.

1.*No argument and no return type:-*

Example:-

```
# include <iostream>
using namespace std;
void prime();     //function declaration
int main()
{
   prime();          //function calling
   return 0;
}
void prime()       //function definition
{
   int a, i, f = 0;
   cout << "Enter a positive integer enter to check: ";
   cin >> a;
   for(i = 2; i <= a/2; ++i)
   {
      if(a% i == 0)
```

```
        {
            f = 1;
            break;
        }
    }
    if (f == 1)
    {
        cout << a<< " is not a prime number.";
    }
    else
    {
        cout <<a << " is a prime number.";
    }
}
```

Output:-

```
Enter a positive integer enter to check: 4  //first time run

4 is not a prime number.

Enter a positive integer enter to check: 5  //second time run

5 is a prime number.
```

2. *No argument but return type:-*

Example:-

```
#include <iostream>
using namespace std;
int prime();        //function declaration
int main()      //main class
{
    int n, i, f = 0;
    n= prime();     //function calling
    for (i = 2; i <= n/2; ++i)
    {
        if (n%i == 0)
        {
            f = 1;
            break;
        }
    }
    if (f == 1)
    {
        cout<<n<<" is not a prime number.";
    }
    else
    {
        cout<<n<<" is a prime number.";
    }
    return 0;
}
int prime()        //function declaration
{
    int n;
    printf("Enter a positive integer to check: ");
    cin >> n;
    return n;
}
```

Output:-

```
Enter a positive integer enter to check: 4  //first time run

4 is not a prime number.
```

```
Enter a positive integer enter to check: 5  //second time run

5 is a prime number.
```

3.*Argument but no return type:-*

Example:-

```
#include <iostream>

using namespace std;

void prime(int n);  //function declaration with argument

int main()

{

    int num;

    cout << "Enter a positive integer to check: ";

    cin >> num;

     prime(num);

    return 0;

}

void prime(int n)

{

    int i, flag = 0;

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

    {

        if (n%i == 0)

        {
            flag = 1;
            break;
        }
    }
    if (flag == 1)
    {
        cout << n << " is not a prime number.";
    }
    else {
        cout << n << " is a prime number.";
    }
}
```

Output:-

```
Enter a positive integer enter to check: 4  //first time run
4 is not a prime number.
Enter a positive integer enter to check: 5  //second time run
5 is a prime number.
```

5.Argument but return type.

6. Example:-

```
#include <iostream>
using namespace std;
int prime(int n);
int main()
```

```
{
    int num, flag = 0;
    cout << "Enter positive integer to check. ";
    cin >> num;
    flag = prime(num);
    if(flag == 1)
        cout << num << " is not a prime number.";
    else
        cout<< num << " is a prime number.";
    return 0;
}
int prime(int n)
{
    int i;
    for(i = 2; i <= n/2; ++i)
    {
        if(n % i == 0)
            return 1;
    }
    return 0;
}
```

Output:-

```
Enter positive integer to check. 4  //first time run

4 is not a prime number.

 Enter positive integer to check. 5  //second time run

5 is a prime number.
```

Created by Ajay Kumar Verma