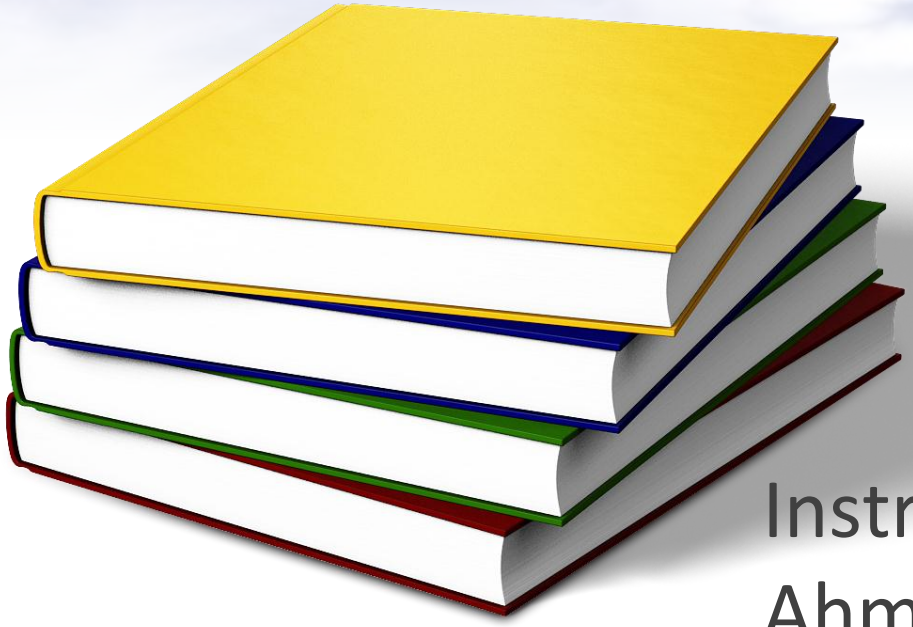
# Chapter # 5
# Functions



Instructor: Engr. Nauman
Ahmad Tariq

# Objectives

- Function declarations & definitions.
- Arguments & return values.
- Reference arguments.
- Overloaded functions.
- Default arguments.

# Function

- Declaration:

```
return_type func_name(func_arguments);
```

- Definition:

```
return_type func_name(func_arguments)
{
//Function body
}
```

- calling Function:

```
func_name(func_arguments);
```

# Example

```
#include<iostream>
#include<conio.h>
using namespace std;
void starline();    //function declaration
int main()
{
starline();          //function call
cout<<“Function”;
starline();          //function call
getche();
return 0;
}
```

# Contd….

```
void starline()   //function definition
{
for(int i=0;i<50;i++)
cout<<"*";
cout<<endl;
}
```

//function body

# Passing Arguments to Function (Constants)

```
#include<iostream>
#include<conio.h>
using namespace std;
Void add(int, int);    //function declaration
int main()
{
add(30,40);          //function call
getche();
return 0;
}
void add(int a, int b)    //function definition
{
cout<<"Addition="<<a+b;
}
```

# Passing Arguments to Function (Variables)

```
#include<iostream>
#include<conio.h>
using namespace std;
Void add(int, int);    //function declaration
int main()
{
int num1,num2;
cout<<"Enter two numbers";
cin>>num1>>num2;
add(num1, num2);          //function call
getche();
return 0;
}
void add(int a, int b)    //function definition
{
cout<<"Addition="<<a+b;
}
```

# Passing Structures as Arguments

```
#include<iostream>
#include<conio.h>
using namespace std;
struct area                 //structure
{
        int length;
        int width;
};
void room(area b)     //function
{
cout<<"area="<<b.length*b.width;
}
```

```
int main()
{
area a;
a.length=10;
a.width=20;
room(a);
getche();
return 0;
}
```

# Returning Values from the Function

```
#include<iostream>
#include<conio.h>
using namespace std;
#define PI 3.14
float circle(int radius)
{
float area;
area=PI*radius*radius;
return area;
}
```

```
int main()
{
int r;
float a;
cout<<"Enter radius:";
cin>>r;
a=circle(r);
cout<<"\nArea of circle="<<a;
getche();
return 0;
}
```

# Returning Structures

```
#include<iostream>
#include<conio.h>
using namespace std;
struct area                    //structure
{
        int length;
        int width;
};
area area_sum(area b,area c)    //function
{
area total ;
total.length= b.length + c.length
total.width=b.width + c.width;
return total;
}
```

```
int main()
{
area a={30,10};
area b={10,20};
area res;
res=area_sum(a,b);
float total_area=res.length*res.width;
cout<<"Total area="<<total_area;
getche();
return 0;
}
```

# Arguments Passed by Value

```
#include<iostream>
#include<conio.h>
using namespace std;
void exchange(int num1, int num2)
{
        int temp;
        temp=num1;
        num1=num2;
        num2=temp;

        cout<<"\nNum1="<<num1;
        cout<<"\nNum2="<<num2;
}
```

```
int main()
{
        int a,b;
        cout<<"Enter two
numbers:";
        cin>>a>>b;
        exchange(a,b);
        cout<<"\na="<<a;
        cout<<"\nb="<<b;

getche();
return 0;
}
```

# Arguments Passed by Reference

```
#include<iostream>
#include<conio.h>
using namespace std;
void exchange(int& num1, int& num2)
{
        int temp;
        temp=num1;
        num1=num2;
        num2=temp;

        cout<<"\nNum1="<<num1;
        cout<<"\nNum2="<<num2;
}
```

```
int main()
{
        int a,b;
        cout<<"Enter two numbers:";
        cin>>a>>b;
        exchange(a,b);
        cout<<"\na="<<a;
        cout<<"\nb="<<b;

getche();
return 0;
}
```

# Overloaded Functions
# Different Number of Arguments

```
#include<iostream>
#include<conio.h>
using namespace std;

void add();
void add(int,int);

int main()
{
add();
int a,b;
cout<<"enter two numbers";
cin>>a>>b
add(a,b);
getche();
return 0;
}
void add()
{
        cout<<"Sum="<<30+20;
}
void add(int a,int b)
{
        cout<<"Sum="<<a+b;
}
```

# Overloaded Functions
# Different Kinds of Arguments

```
#include<conio.h>
using namespace std;

void add(float, float);
void add(int,int);

int main()
{
add(3.14, 5.34);
int a,b;
cout<<"enter two numbers";
cin>>a>>b
add(a,b);
getche();
return 0;
}
void add(float f, float m)
{
        cout<<"Sum="<<f+m;
}
void add(int a,int b)
{
        cout<<"Sum="<<a+b;
}
```

# Default Arguments

```
#include<iostream>
#include<conio.h>
using namespace std;
void repchar(char='*',int=40);
int main()
{
repchar();
repchar('-');
repchar('=',50);
getche();
return 0;
}
void repchar(char ch, int n)
{
        for(int i=0;i<n;i++)
                cout<<ch;
        cout<<endl;
}
```

# Variables

| | Automatic (Local) | Static | External |
|---|---|---|---|
| Visibility | function | function | file |
| Lifetime | function | program | program |
| Initialized value | Not initialized | 0 | 0 |
| Purpose | Variables used by a single function | Same as auto but must retain value when function terminates | Variable used by several functions |

# Returning By reference

```
#include<iostream>
#include<conio.h>
using namespace std;
int x;
int& setx();
int main()
{
setx()=100;
cout<<"x="<<x<<endl;
getche();
return 0;
}

int& setx()
{
        return x;
}
```