Object Oriented Programming in C++

Nepal College of Information Technology

Course Instructor: Er. Rabina Chaudhary

Passing Object as Argument:

- An object can be passed to a function as an argument like normal variables
- When an object of a class is passed to a function, all data members of the class are passed in function

Syntax: Declaration of function with object as argument

```
return_type functionName(className object_name);
```

calling the function:

```
object.functionName(object_name);
```

Passing Integer as Argument:

While passing integer as argument in function

function declaration:

```
void testFunction(int x, int y );
```

function call:

```
testFunction(a,b);//a and b are integer variables
```

Passing Object as Argument:

While passing object as argument in function

function declaration:

```
void exampleFunction( class_name obj,class_name obj2);
```

function call:

```
object.exampleFunction(objectX,objectY);//objectX
//and  objectY are objects
```

Q. Define a class named Complex with data members real and img. Use appropriate member function of the class which accepts two objects of the class and add them.

```
#include<iostream.h>
#include<conio.h>
class Complex
{
        private:
        int real,img;
  public:
        void getData();
        void addComplex(Complex C1, Complex C2);
         void display();
};
void Complex::getData()
{
        cout<<endl<<"Real part : ";
        cin>>real;
        cout<<"imaginary part :";
        cin>>img;
}
void Complex::display()
{
if(img>0)
        cout<<real<<"+j"<<img;

  else
        cout<<real<<"-j"<<(-1)*img;
}
```

Program continued

```
void Complex::addComplex(Complex C1,Complex  C2)
{
        real=C1.real+C2.real;
        img=C1.img+C2.img;
}
```

```
void main()
{
  Complex first,second,result;
  cout<<"Enter first complex number";
  first.getData();
  cout<<"Enter second complex number";
  second.getData();
  cout<<"First complex number is :";
  first.display();
  cout<<endl<<"Second complex number is :";
  second.display();
  cout<<endl<<" The sum is : "<<endl;
  result.addComplex(first,second);
  result.display();
  getch();
}
```

Practice:

Q1. Write a program to define class Distance with data members feet and inches of appropriate type. Define member function of the class which accepts two objects of the class and adds them.

Q2. Write a program to define a class named Time with data members hours, minutes and seconds. Use appropriate member function of the class which accepts two objects of the class and adds them.

Q2. Write a program to define a class named Time with data members hours, minutes and seconds. Use appropriate member function of the class which accepts two objects of the class and add them.

```
#include<conio.h>
#include<iostream.h>
class Time
{
private:
        int hour,minute,second;
  public:
        void setTime();
         void displayTime();
         void addTime(Time, Time);
};
```

Program Continued…

```
void Time::setTime()
{
   cout<<"Enter hour :";
   cin>>hour;
   cout<<"Enter minute :";
   cin>>minute;
   cout<<"Enter second :";
   cin>>second;
}
```

Program Continued…

```
void Time::displayTime()
{
cout<<hour<<":"<<minute<<":"<<second<<endl;
}


void Time::addTime(Time  t1,Time t2)
{
  second=t1.second+t2.second;
  minute=second/60;
  second=second%60;

  minute=minute+t1.minute+t2.minute;
  hour=minute/60;
  minute=minute%60;

  hour=hour+t1.hour+t2.hour;
}
```

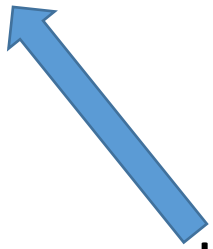
Program Continued…

```
void main()
{
Time time1,time2,time3;
  cout<<"Enter first time"<<endl;
  time1.setTime();
  cout<<endl<<"Enter second time"<<endl;
  time2.setTime();
  cout<<endl<<"The first time is : ";
  time1.displayTime();
  cout<<endl<<"The second time is : ";
  time2.displayTime();

  time3.addTime(time1,time2);

  cout<<endl<<"The resultant time after addition is : ";
  time3.displayTime();

  getch();
}
```

Returning Integer from function:

Syntax:

```
int function_name(arguments);
```



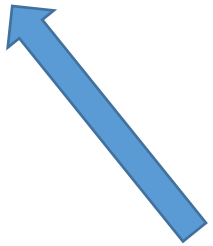
return_type

Returning Object from function:

A function can return object to the calling function as it can return integer, float or character values.

Syntax:

```
class_name function_name(arguments);
```



return_type

Example:

Q1. Write a program to define class Complex with data members real and img. Define a member function to add two complex numbers and return the sum of them.

```
#include<iostream.h>
#include<conio.h>

class Complex
{
        private:
        int real,img;
  public:
        void getData();
        Complex addComplex(Complex C2);
        void display();
};
```

Program continued…

```
void Complex::getData()
{
        cout<<endl<<"Real part : ";
        cin>>real;
        cout<<"imaginary part :";
        cin>>img;
}
void Complex::display()
{
        if(img>0)
        {
                cout<<real<<"+j"<<img;
        }
        else
        {
        cout<<real<<"-j"<<(-1)*img;
        }
}
```

Program continued…

```
Complex Complex::addComplex(Complex   C2)
{
        Complex C3;
        C3.real=real+C2.real;
        C3.img=img+C2.img;
        return C3;
}
```

Program continued…

```
void main()
{
                Complex first,second,result;
                 cout<<"Enter  first complex number";
                first.getData();
                cout<<"Enter  second complex number";
                second.getData();
                cout<<"First complex number is :";
                first.display();
                cout<<endl<<"Second  complex number is :";
                second.display();
                cout<<endl<<"  The sum is : "<<endl;
                result=first.addComplex(second);
                result.display();
  getch();
}
```

Practice:

Q2. Write a program to define class Distance with data members feet and inches of appropriate type. Define member function of the class to adds two distances and return the sum in main program.

Q3. Write a program to define a class named Time with data members hours, minutes and seconds. Use appropriate member function of the class which accepts two objects of the class and adds them and returns the result.