Examples of Classes

```
#include <iostream>
using namespace std;
class Cat {
 private:
      int age;
    int weight;
  public:
    char gender;
    void setAge (int yrs) { age = yrs; }
    void setWeight (int kgs) { weight = kgs; }
    int getAge() { return age; }
    int getWeight() { return weight; }
 };

int main( )
{
    Cat MyCat;
    MyCat.setAge(3);
    MyCat.setWeight(2);
    MyCat.gender='m';
    cout << "My cat‘s age is " << MyCat.getAge();
    cout << " and weighs " << MyCat.getWeight() << "kg\n";
    cout << "and its gender is "<< MyCat.gender<<endl;
    return 0;
    }
```

OUTPUT:

```
My cat‘s age is 3 and weighs 2kg
and its gender is m
```

```
#include <iostream>
#include <string>
using namespace std;
class Course {
    // Data Member
public:
    string studentName;
    string courseCode ;
};

int main() {
Course course1, course2;
// assign values to course1
course1.courseCode= "CSC1201";
course1.studentName= "Muna AlKebir";
//assign values to course2
course2.courseCode= "csc1301";
course2.studentName= "Salwa AlAmri";
cout<< course1.studentName << " has the course "<<course1.courseCode<<
endl;
cout<<course2.studentName << " has the course "<<course2.courseCode<<
endl;
    return 0;
}
```

OUTPUT:

```
Muna AlKebir has the course CSC1201
Salwa AlAmri has the course csc1301
```

```cpp
#include <iostream>
using namespace std;
class Circle
{
private:
float radius;

public:  //prototype only !
// constructors
Circle()
    {radius=0;}
Circle(float r)
    {setRadius(r);}
// destructor
~Circle()
    {cout<<" ending object...\n";}
void setRadius(float r)
    {

        if ( r>=0.0)
            radius=r;
        else
            radius=0.0;
}
float getRadius(){return radius;}
float area(){return 3.14*radius*radius;}
float perimeter(){return 2 * 3.14 * radius;}
};

int main()
{
 float x;
cout<<"Enter the radius of the circle: ";
    cin>>x;
    Circle C1(x);
cout<<"the area of the circle is: "<<C1.area()<<endl;
cout<<" and the perimeter is:"<<C1.perimeter()<<endl;

    return 0;
}
```

OUTPUT:

```
Enter the radius of the circle: 4
the area of the circle is: 50.24
 and the perimeter is:25.12
 ending object...
```

```cpp
#include <iostream>
using namespace std;
class queue {
    int q[100];
    int sloc, rloc;
public:
    queue();
    void qput (int i);
    int qget ();
    };
queue::queue()
{rloc=sloc=0;}
void queue::qput(int i)
{
    if(sloc==100)
    {cout << " Queue is full.\n";
        return;
    }
    sloc++;
    q[sloc]=i;
}
  int queue::qget ()
{
    if(rloc == sloc)
    {
        cout << "Queue underflow.\n";
        return 0;
    }
    rloc++;
    return q[rloc];
          }

int main ()
{
    queue a, b;
    a.qput (10);
    b.qput (19);
    a.qput (20);
    b.qput (1);
    cout << "Contents of queue a: ";
    cout << a.qget() << " ";
    cout << a.qget() << " \n";
    cout << "Contents of queue b: ";
    cout << b.qget () << " ";
    cout << b.qget () << " \n";
    return 0;
}
```

OUTPUT:

```
Contents of queue a: 10 20
Contents of queue b: 19 1
```

```
#include <iostream>
using namespace std;
class myclass {
    int a; // private data
public:
    int b; // public data
    void setab (int i); // public functions
    int geta () ;
    void reset();
};
void myclass::setab(int i)
{
    a = i; // refer directly to a
    b = i*i; // refer directly to b
}
int myclass::geta (){ return a;} // refer directly to a
void myclass::reset (){ // call setab() directly
    setab (0); // the object is already known
}
int main ()
{
    myclass ob;
    ob.setab (5); // set ob.a and ob.b
    cout<< "ob after setab (5): ";
    cout<<ob.geta()<<' ';
    cout << ob.b; // can access b because it is public
    cout <<'\n';
    ob.b = 20; // can access b because it is public
    cout << "ob after ob.b=20: ";
    cout << ob.geta()<<' ';
    cout << ob.b<<endl;
    ob.reset ();
    cout << "ob after ob.reset (): ";
    cout << ob.geta () << ' ';
    cout << ob.b<<endl;
    return 0;
}
```

OUTPUT:

```
ob after setab (5): 5 25
ob after ob.b=20: 5 20
ob after ob.reset (): 0 0
```

```
#include <iostream>
using namespace std;

class queue
{
    int q[100];
    int sloc, rloc;
public:
    queue();
    ~queue();
    void qput(int i);
    int qget (); };

    queue::queue()
      {
          rloc=sloc=0;
          cout<<" Queue initialized. \n";
      }
   queue::~queue()
        {

            cout<<" Queue destroyed. \n";
        }
    void queue::qput(int i)
    {
        if(sloc==100)
        {
            cout << " Queue is full.\n";
            return;
        }
        sloc++;
        q[sloc]=i;
    }
      int queue::qget ()
    {
        if(rloc == sloc)
        {
            cout << "Queue underflow.\n";
            return 0;
        }
        rloc++;
        return q[rloc];
                 }
```

```
int main ()
{
    {
    queue a;
    a.qput (10);
    a.qput (20);
    cout << a.qget() << " ";
    cout << a.qget() << " \n";
}
    {

     queue b;
    b.qput (19);
    b.qput (1);
    cout << b.qget() << " ";
    cout << b.qget() << " \n";
    }
    cout<<"End of main"<<endl;
    return 0;
}
```

OUTPUT:

```
Queue initialized.
10 20
 Queue destroyed.
 Queue initialized.
19 1
 Queue destroyed.
End of main
```