Lab No (7)

7.1 **Execute the following code and show the output:**

```
class A
{  public:
    virtual void f() const { std::cout << "base"; } };
class B: public A
{  public:
    virtual void f() const { std::cout << "derived"; } };
std::ostream& operator<<(std::ostream& os, const A& a )
{       a.f();
     return os; }
int main()
{      B b;
    std::cout << b << std::endl;

    return 0;  }
```

7.2 **Execute the following code and show the output:**

```
class Weapon
{      public:
       void loadFeatures()
         { cout << "Loading weapon features.\n"; } };
class Bomb : public Weapon
{      public:
       void loadFeatures()
         { cout << "Loading bomb features.\n"; } };
class Gun : public Weapon
{      public:
       void loadFeatures()
         { cout << "Loading gun features.\n"; } };
int main()
{  Weapon *w = new Weapon;
    Bomb *b = new Bomb;
    Gun *g = new Gun;
    w->loadfeatures();
    b->loadFeatures();
    g->loadFeatures();
    return 0; }
```

7.3 Write a program that has an abstract base class named **Quad**. This class should have four member data variables (floats) representing side lenghts and a pure virtual function **Area**. It should also have a method for setting the data variables. Derive a class **Rectangle** from **Quad** and override the **Area** method so that it returns the area of the **Rectangle**. Write a main function that creates a **Rectangle** and sets the side lengths. Also write a top-level function that will take a parameter of type **Quad** and return the value of the appropriate **Area** function.

7.4 Write a class that contains two class data members **numBorn** and **numLiving**. The value of **numBorn** should be equal to the number of objects of the class that have been instanced. The value of **numLiving** should be equal to the total number of objects in existence currently (*ie, the objects that have been constructed but not yet destructed*.)