DRY-RUN EXERCISES

8.1 What will happen when the following program is run?

```
#include<iostream>
using namespace std;
int a,b,c,d,e;

class A
{
      protected:
      public:
      void value_a_b()
      {
            a=2; b=2;
      }
};

class B:public A
{
      protected:
      public:
      void value_c()
      {
            c=2;
      }
};
class C
{
      protected:
      public:
      void value_d()
      {
           d=3;
      }
};
class D:public B, public C
{
      protected:
      public:
   void result()
      {
           value_a_b();
           value_c();
```

```
          value_d();
          e=a*b*c*d;
          cout<<"\n Multiplication is :"<<e;
      }
};
int main()
{
    D d1;
    d1.result();
    return 0;
}
```

8.2 What will happen when the following program is run?

```
#include<iostream>
using namespace std;
class B
{
  int x;
  public:
  B() { cout << "1\t"; }
};
class D : public B
{
  int y;
  public:
  D() { cout << "2\t"; }
  D(int i) { cout << "3\t"; }
};
int main()
{
 B b;
 D d1;
 D d2(1);
}
```

8.3 What will happen when the following program is run?

```
#include <iostream>
using namespace std;
class A
{
 int a;
 public:
 A() {}
 A(int d)
 {  a = d; }
 ~A()
 {
  cout << endl << "Object A being destroyed";
 }
 void show()
 {
  cout << a << endl;
 }
};
class B : public A
{
 int b;
 public:
 B() : A()
 {
 }
 B(int d, int d1) : A(d)
 {
  b = d1;
 }
 ~B()
 {
  cout << endl << "Object B is being destroyed";
 }
 void show()
 {
  A::show();
  cout << b << endl;
 }
};
int main()
{
 B *p;
 p = new B(20, 30);
 (*p).show();
 delete p;
 return 0;
}
```

8.4 What will happen when the following program is run?

```
#include <iostream>
using namespace std;
class A
{
 int a;
 public:
 A(){}
 A(int d) {  a = d;  }
 ~A()
 {
  cout << "OBject of A being deleted" << endl;
 }
 void show_A()
 {
  cout << a << endl;
 }
};
class B
{
 int b;
 public:
 B(){}
 B(int d)
 {
  b = d;
 }
 ~B()
 {
  cout << “Object of B being deleted” << endl;
 }
 void show_B()
 {
  cout << b << endl;
 }
};
class C: public A, public B
{
 int c;
 public:
 C(): A(), B() { }
 C(int d1, int d2, int d3) : A(d1), B(d2)
 {
  c = d3;
 }
 ~C()
 {
  cout << “Object of C being deleted” << endl;
 }
 void show_C()
 {
  show_A();
  show_B();
  cout << c << endl;
 }
};

int main()
{
  C *p;
  p = new C(10, 20, 30);
  p -> show_C();
  delete p;
  return 0;
}
```