NIKHIL RAJ

9921103016 – F5

SDF TUT- 6

1-D

2-C

3-B

4-B

5-Constructor can't be virtual as when a constructor of a class is executed there is no virtual table in the memory means no virtual pointer defined yet.

(B)

6-A

7-A

8-B

9-A

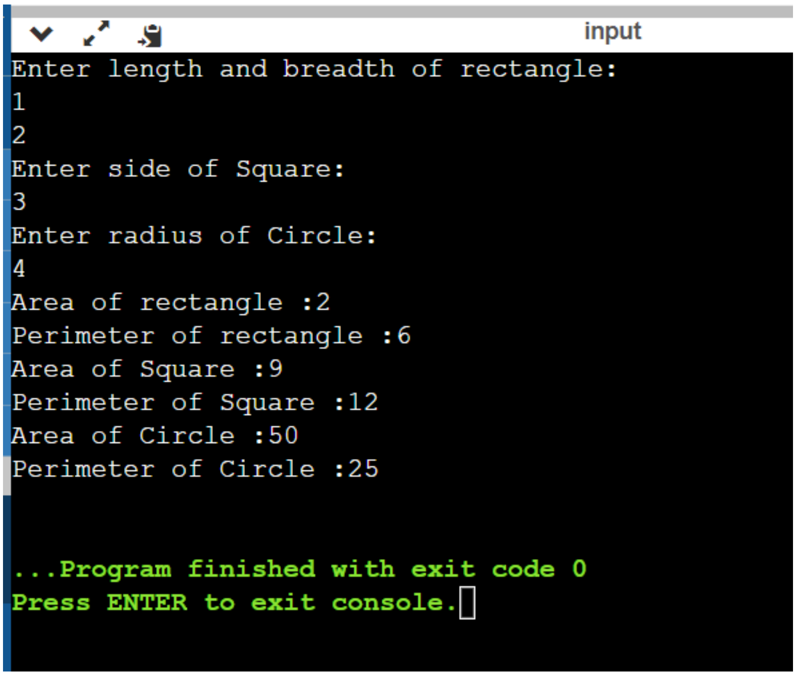
10-

```
#include <iostream>
using namespace std;
class shape
{
  public:
  int a,p;
  virtual void area()=0;
  virtual void perimeter()=0;
};
class rectangle : public shape
```

```
{
    public:
    int rl,rb;
    void set_data(int l,int b)
  {
      rl=l;
      rb=b;
  }
    void area()
    {
        a=rl*rb;
        cout<<"Area of rectangle :"<<a<<endl;
    }
    void perimeter()
    {
        p=2*(rl+rb);
        cout<<"Perimeter of rectangle :"<<p<<endl;
    }
};
class square : public shape
{
    public:
    int ss;
    void set_data(int s)
  {
      ss=s;
  }
```

```
   void area()
   {
      a=ss*ss;
      cout<<"Area of Square :"<<a<<endl;
   }
   void perimeter()
   {
      p=4*ss;
      cout<<"Perimeter of Square :"<<p<<endl;
   }
};
class circle : public shape
{
   public:
   int cr;
   void set_data(int r)
  {
    cr=r;
  }
   void area()
   {
      a=3.14*cr*cr;;
      cout<<"Area of Circle :"<<a<<endl;
   }
   void perimeter()
   {
      p=2*3.14*cr;
```

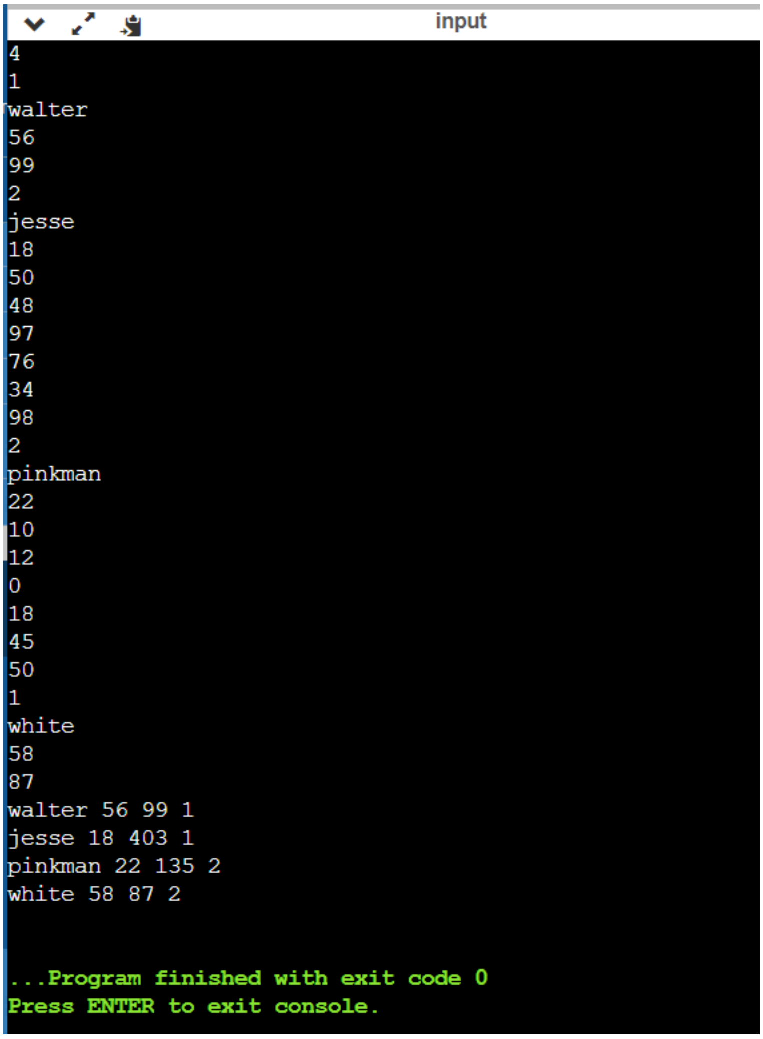
```
        cout<<"Perimeter of Circle :"<<p<<endl;
    }
};
int main()
{
    shape *h;
    rectangle r1;
    square s1;
    circle c1;
    int p,q,m,n;
    cout<<"Enter length and breadth of rectangle: \n";
    cin>>p>>q;
    cout<<"Enter side of Square: \n";
    cin>>m;
    cout<<"Enter radius of Circle: \n";
    cin>>n;

    h=&r1;
    r1.set_data(p,q);
    h->area();
    h->perimeter();

    h=&s1;
    s1.set_data(m);
    h->area();
    h->perimeter();
    h=&c1;

    c1.set_data(n);

    h->area();

    h->perimeter();


    return 0;

}
```



```
Enter length and breadth of rectangle:
1
2
Enter side of Square:
3
Enter radius of Circle:
4
Area of rectangle :2
Perimeter of rectangle :6
Area of Square :9
Perimeter of Square :12
Area of Circle :50
Perimeter of Circle :25


...Program finished with exit code 0
Press ENTER to exit console.
```

11-

```
#include <iostream>

#include <algorithm>

using namespace std;


class Person {

protected:

    std::string name;

    int age;
    int cur_id;
public:
    Person()
        : name(""), age(0), cur_id(0)
    {   }

    virtual void getdata() { cin >> name >> age; }
    virtual void putdata() { cout << name << " " << age << endl; }
};

class Professor : public Person {
    int publications;
    static int id;
public:
    Professor()
        : publications(0)
    { cur_id = ++id;  }

    void getdata() override { cin >> name >> age >> publications; }
    void putdata() override { cout << name << " " << age << " " << publications
                              << " " << cur_id << endl; }
};
int Professor::id = 0;

class Student : public Person {
    int marks[6];
    static int id;
    int marksSum;
public:
    Student()
        : marks{0}, marksSum(0)
    { cur_id = ++id; }

    void getdata() override {
        cin >> name >> age;
        for (int i=0; i < 6; i++) {
            cin >> marks[i];
        }
    }
    void putdata() override {
        for(int i : marks)
            marksSum += i;

        cout << name << " " << age << " " << marksSum << " " << cur_id << endl;
    }
};
int Student::id = 0;

int main(){

    int n, val;
    cin>>n; //The number of objects that is going to be created.
    Person *per[n];
    for(int i = 0;i < n;i++){


        cin>>val;

        if(val == 1){

            // If val is 1 current object is of type Professor

            per[i] = new Professor;


        }

        else per[i] = new Student; // Else the current object is of type Student


        per[i]->getdata(); // Get the data from the user.


    }


    for(int i=0;i<n;i++)

        per[i]->putdata(); // Print the required output for each object.


    return 0;


}
```



```
4
1
walter
56
99
2
jesse
18
50
48
97
76
34
98
2
pinkman
22
10
12
0
18
45
50
1
white
58
87
walter 56 99 1
jesse 18 403 1
pinkman 22 135 2
white 58 87 2


...Program finished with exit code 0
Press ENTER to exit console.
```

12-

A virtual destructor is used to free up the memory space allocated by the derived class object or instance while deleting instances of the derived class using a base class pointer object.