
FACE

Inheritance

- The capability of a class to derive properties and characteristics from another class.

- Syntax:

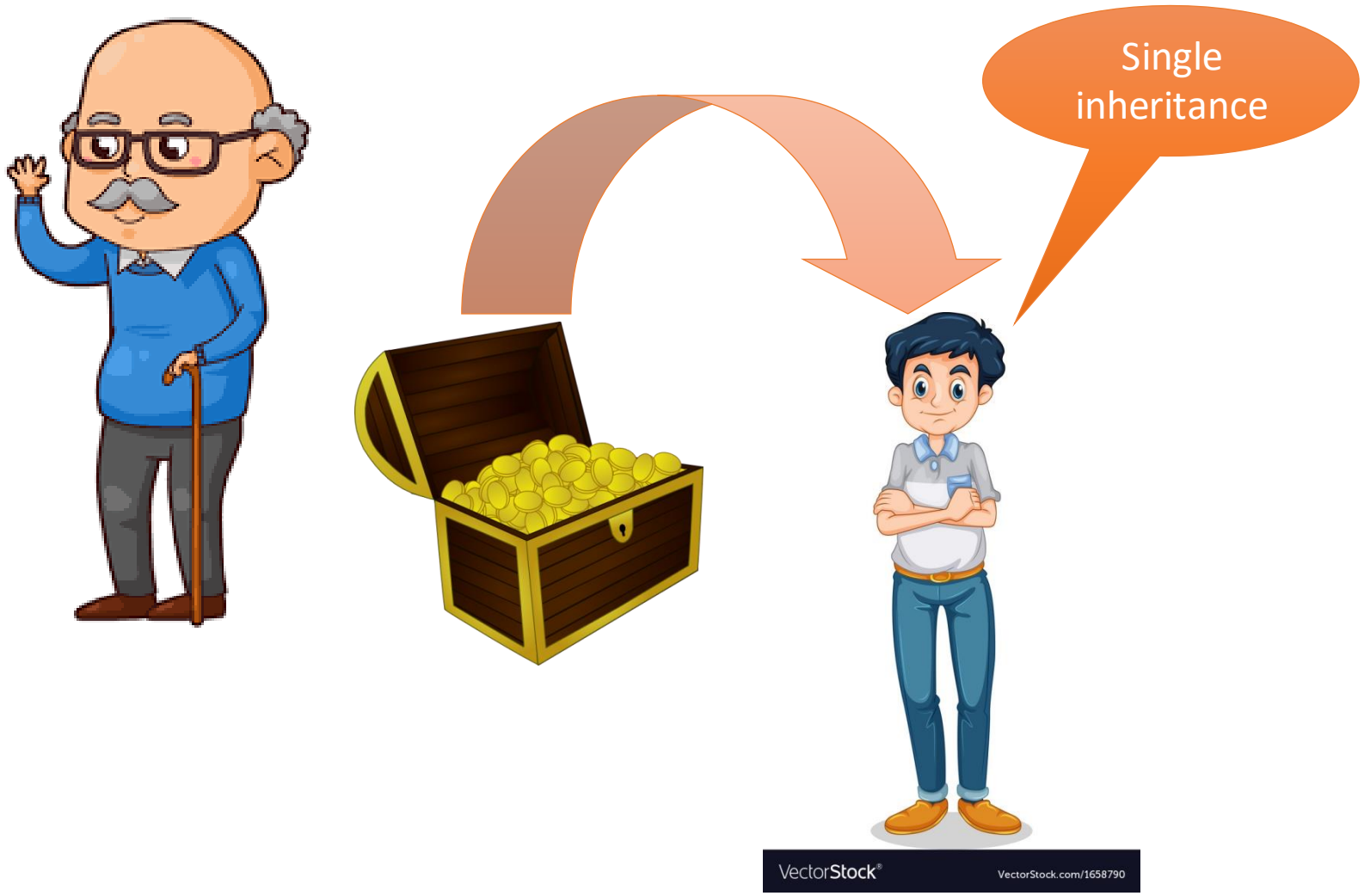
```
class derived-class : access-specifier base-class
{
    .... ....
}
```

FACE

Types of Inheritance

1. Single Inheritance
2. Multiple Inheritance
3. Hierarchical Inheritance
4. Multilevel Inheritance
5. Hybrid Inheritance (also known as Virtual Inheritance)

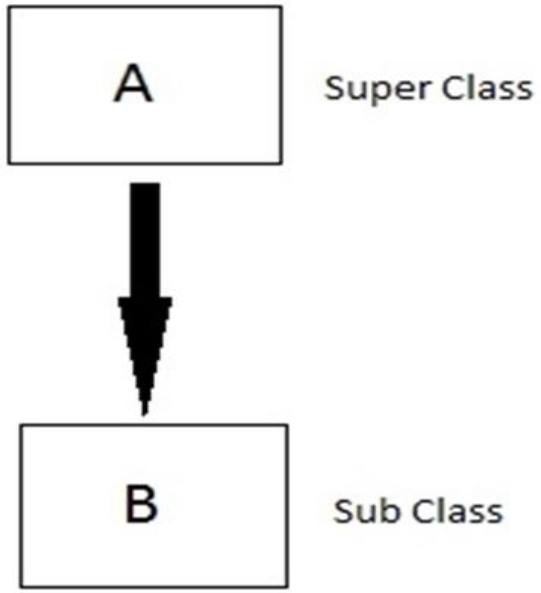
FACE



FACE

1. Single Inheritance

- one derived class inherits from only one base class



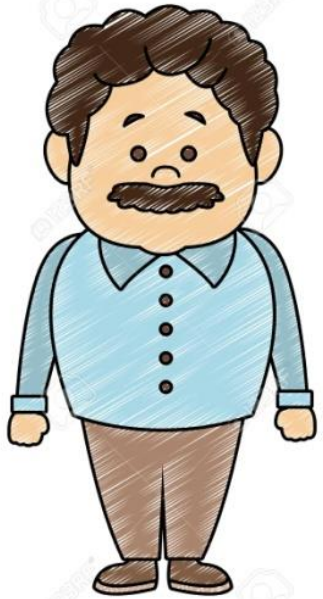
FACE

```
1  #include <iostream>
2  using namespace std;
3  class  Shape
4  {
5      protected:
6        int width;
7        int height;
8      public:
9        void  set_values(int w, int h)
10        {
11            width = w;  height = h;
12        }
13 };
14 class Rectangle: public Shape
15 {
16    public:
17       int getArea()
18       {
19           return (width * height);
20       }
21 };
22
```

```
1   int main()
2   {
3       Rectangle Rect;
4       Rect.set_values (5, 3);
5       cout<<"Total area: "<<Rect.getArea();
6       return 0;
7   }
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
```

OUTPUT

```
Total area: 15
```

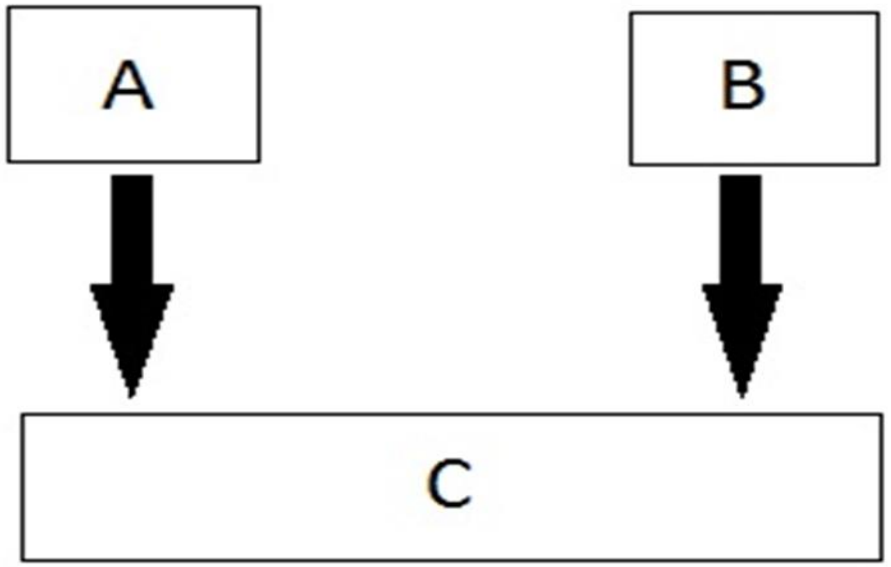


Inheriting their properties!!

FACE

2. Multiple Inheritance

- single derived class may inherit from two or more than two base classes.



FACE

```
1   #include <iostream>
2   using namespace std;
3   class Student
4   {
5         protected:
6             int roll, m1, m2;
7         public:
8            void get()
9            {
10               cout << "Enter the Roll No: ";
11               cin >> roll;
12               cout << "Enter the marks:   ";
13               cin >> m1 >> m2;
14           }
15  };
16
17
18
19
20
21
22
```

```
1  class Extracurriculum
2  {
3    public:
4      int xm;
5    public:
6      void get_xm()
7      {
8         cout<<"Enter the Extra curriculum mark ";
9         cin >> xm;
10     }
11 };
12 class Average : public Student, public Extracurriculum
13 {
14          float total;
15          public:
16          void display()
17          {
18               total = (m1 + m2 + xm);
19               cout << "\nRoll No: "<<roll<< "\nTotal: "<<total;
20          }
21 };
22
```

```
1   int main()
2   {
3       Average A1;
4       A1.get();
5       A1.get_xm();
6       A1.display();
7       return 0;
8   }
9
10
11
12
13
14
15
16
17
18
19
20
21
22
```

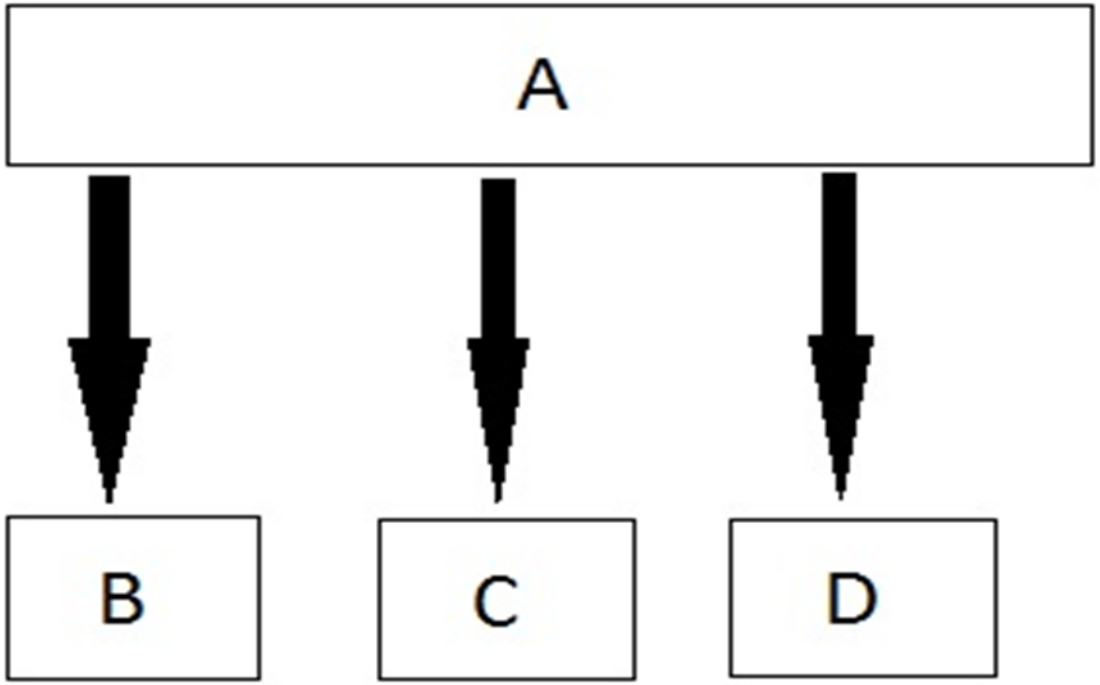
OUTPUT

```
Enter the Roll No: 131439
Enter the marks:   89  89
Enter the Extra curriculum mark 90

Roll No: 131439
Total: 268
Average: 89.3333
```

3. Hierarchical Inheritance

- multiple derived classes inherits from a single base class.



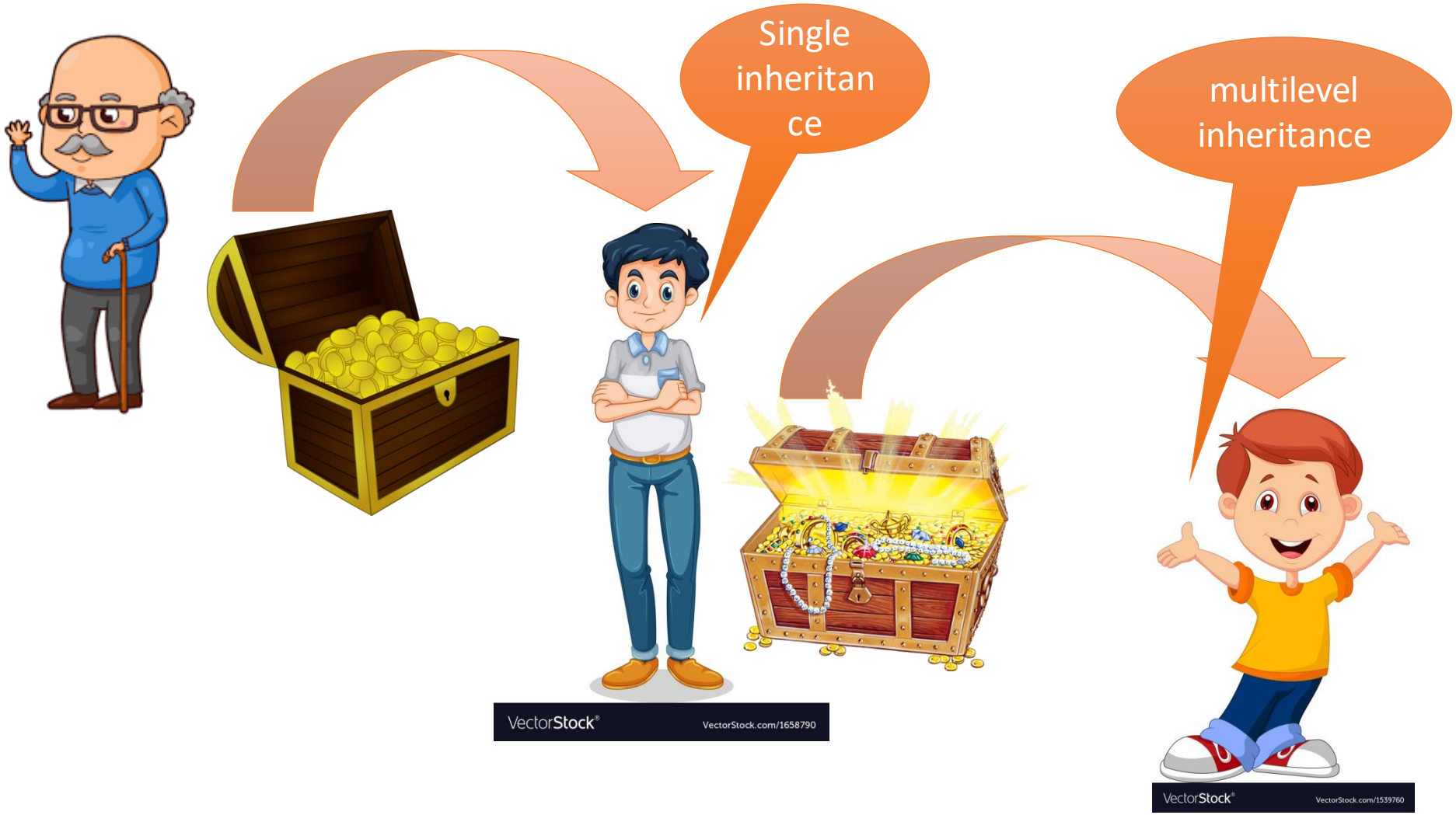
FACE

```
1  #include <iostream>
2  using namespace std;
3  class A
4  {
5      public:
6        int x, y;
7        void getdata()
8        {
9            cout << "Enter value of x and y:\n";
10           cin >> x >> y;
11       }
12 };
13 class B : public A
14 {
15     public:
16       void product()
17       {
18           cout << "Product= " << x * y<<endl;
19       }
20 };
21
22
```

```
1  class C : public A
2  {
3      public:
4        void sum()
5        {
6          cout << "Sum= " << x + y;
7        }
8  };
9  int main()
10 {
11     B obj1;
12     C obj2;
13     obj1.getdata();
14     obj1.product();
15     obj2.getdata();
16     obj2.sum();
17     return 0;
18 }
19
20
21
22
```

OUTPUT

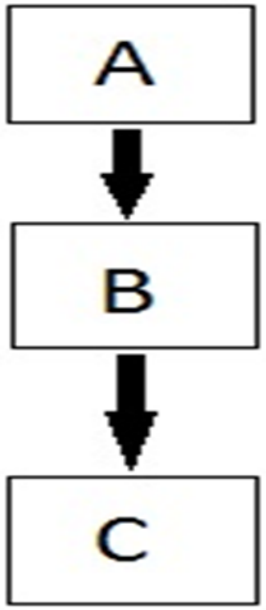
```
Enter value of x and y:
10
20
Product= 200
Enter value of x and y:
30
40
Sum= 70
```


Single inheritan ce
multilevel inheritance
VectorStock®
VectorStock.com/1658790
VectorStock®
VectorStock.com/1539760

FACE

4.Multilevel Inheritance

- the derived class inherits from a class, which in turn inherits from some other class



FACE

```
1  #include <iostream>
2  using namespace std;
3  class Base
4  {
5      public:
6      void display1()
7      {
8        cout<<"Base class content.\n";
9      }
10 };
11 class Derived1 : public Base
12 {
13     public:
14     void display2()
15     {
16       cout<<"1st derived class content.\n";
17     }
18 };
19
20
21
22
```

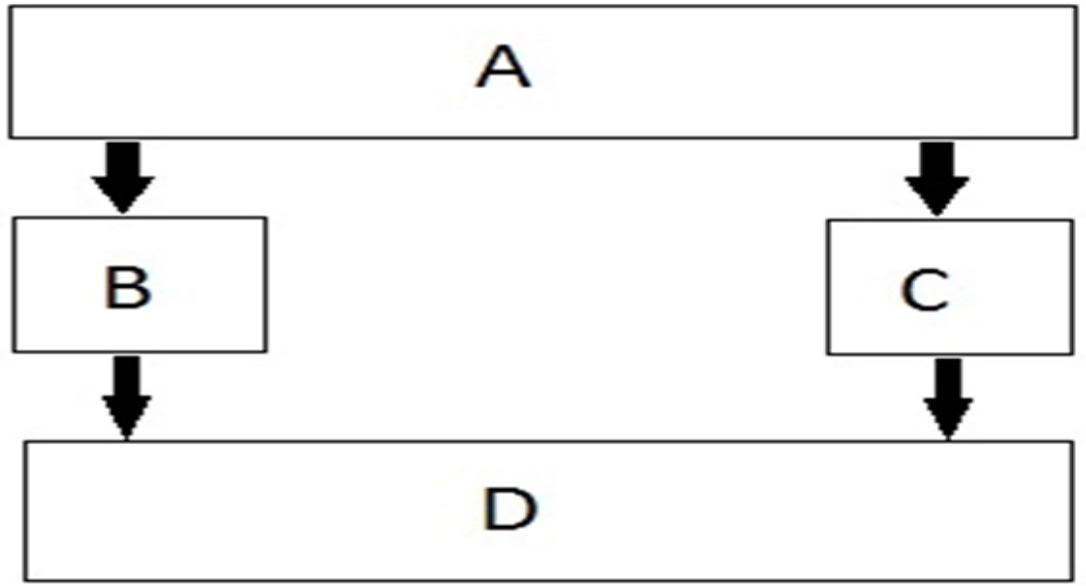
```
1  class Derived2 : public Derived1
2  {
3      public:
4      void display3()
5      {
6          cout<<"2nd Derived class content.\n";
7      }
8  };
9  int main()
10 {
11     Derived2 D;
12     D.display3();
13     D.display2();
14     D.display1();
15 }
16
17
18
19
20
21
22
```

OUTPUT

```
2nd Derived class content.
1st derived class content.
Base class content.
```

5. Hybrid Inheritance

- implemented by combining more than one type of inheritance



FACE

```
1  #include<iostream>
2  using namespace std;
3  class stu
4  {
5        int id;
6        char name[20];
7        public:
8        void getstu()
9        {
10             cout << "Enter stuid and name\n";
11             cin >> id >> name;
12       }
13 };
14
15
16
17
18
19
20
21
22
```

```
1  class marks: public stu
2  {
3        protected:
4        int m, p, c;
5        public:
6        void getmarks()
7        {
8              cout << "Enter 3 subject marks:\n";
9              cin >> m >> p >> c;
10       }
11 };
12 class sports
13 {
14       protected:
15       int spmarks;
16       public:
17       void getsports()
18       {
19             cout << "Enter sports marks:\n";
20             cin >> spmarks;
21       }
22 };
```

```
1  class result : public marks, public sports
2  {
3        int tot;
4        float avg;
5        public :
6        void show()
7        {
8              tot=m+p+c;
9              avg=tot/3.0;
10             cout << "Total=" << tot << endl<< "Average=" << avg << endl;
11             cout << "Average + Sports marks =" << avg+spmarks;
12       }
13 };
14 int main()
15 {
16       result r;
17       r.getstu();
18       r.getmarks();
19       r.getsports();
20       r.show();
21       return 0;
22 }
```

OUTPUT

```
131439
Sanjana
Enter 3 subject marks:
78
89
65
Enter sports marks:
56
Total=232
Average=77.3333
Average + Sports marks =133.333
```

Question 1

_____ have the return type void?

A) Constructor

B) Destructor

C) All functions

D) none

Question 2

How many basic types of inheritance are provided as OOP feature?

A) 4

B) 3

C) 2

D) 1

Question 3

What is the output of the following code?

```
#include <iostream>
using namespace std;
class Face
{
    public:
    Face()
    {
        cout <<"Hi from Face. ";
    }
} f;
```

```
int main()
{
    cout << "You are in Main";
    return 0;
}
```

FACE

Question 3

A) Hi from Face.

B) You are in Main

C) Hi from Face. You are in Main

D) You are in MainHi from Face.

Question 4

What is the output for the following?

```
#include<iostream>
using namespace std;
class base
{
    int arr[10];
};
class b1: public base { };
class b2: public base { };
```

```
class derived: public b1, public
b2 { };
int main()
{
  cout<<sizeof(derived);
  return 0;
}
```

A) 40

B) 60

C) 80

D) 100

Question 5

Which among the following best describes the Inheritance?

A) Copying the code already written

B) Using the code already written once

C) Using already defined functions in programming language

D) Using the data and functions into derived segment

FACE

THANK YOU

FACE