EXP.NO: 3.3

AIM: Write a c++ program to incorporate various forms of Inheritance

i) Single Inheritance:

Program:

```
#include<iostream>
using namespace std;
 class A {
    protected:
        void showa() {
            cout << "class A show()" << endl;
        }
};
class B : public A {
    public:
        void showb() {
            showa();
            cout << "class B show()" << endl;
        }
};

int main() {
    B b;

    b.showb();
    return 0;
}
```

OUTPUT:

```
/Users/yaswanthkakarla/CLionProjects/untitled5/cmake-build-debug/untitled5
class A show()
class B show()

Process finished with exit code 0
```

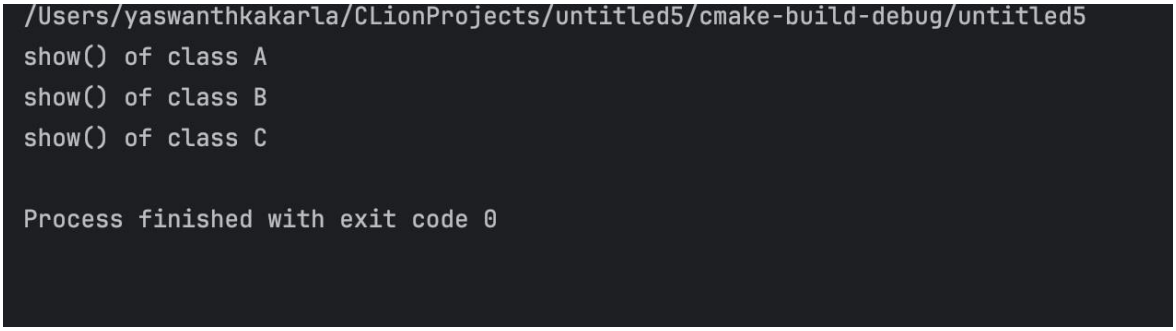
ii) Multiple Inheritance:

Program:

```
#include<iostream>
using namespace std;
class A {
   protected:
      void showa() {
         cout << "show() of class A" << endl;
      }
};
class B {
   protected:
      void showb() {
         cout << "show() of class B" << endl;
      }
};
class C : public A, public B {
   public:
      void showc() {
         showa();

         showb();

         cout << "show() of class C" << endl;
      }
```

```
};

int main() {
    C c;

    c.showc();
    return 0;
}
```

OUTPUT:



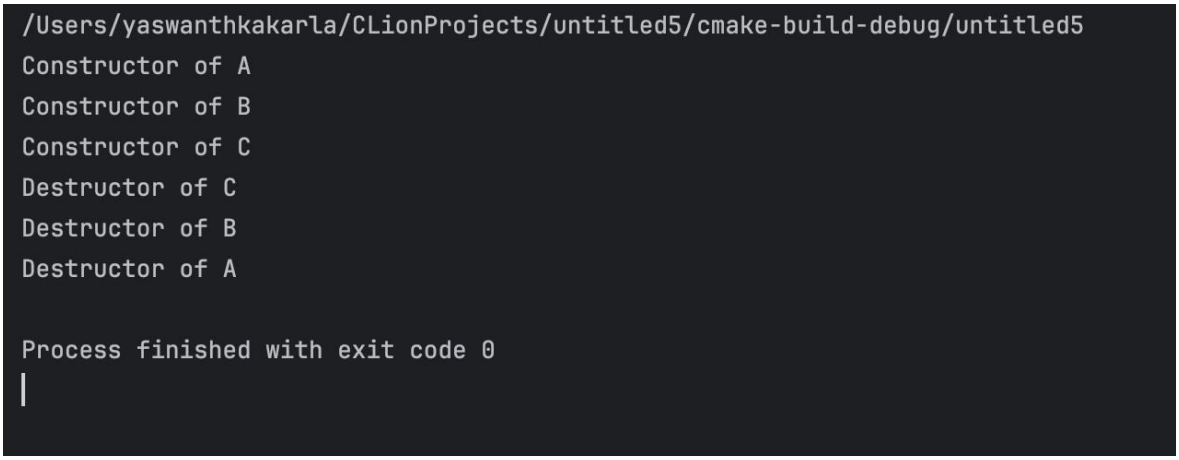
iii) Multi-level Inheritance:

Program:

```
#include<iostream>
using namespace std;
class A {
   public:
A() {
        cout << " Member function of A" << endl;
      }
 ~A() {
        cout << " Member function of A" << endl;
      }
};
class B : public A {
   public:
B() {
        cout << " Member function of B" << endl;
      }
  ~B() {
        cout << " Member function of B" << endl;
      }
};
class C : public B {
   public:
```

```
 C() {
        cout << " Member function of C" << endl;
    }
    ~C() {
        cout << " Member function of C" << endl;
    }
};

int main() {
    C c;

    return 0;
}
```

OUTPUT:



iv) Hierarchical Inheritance:

Program:

```
#include<iostream>
using namespace std;
class A {
   public:
A() {
        cout << " Member function of A" << endl;
      }
      ~A() {
        cout << " Member function of A" << endl;
      }
};
class B : public A {
   public:
      B() {
        cout << " Member function of B" << endl;
      }
      ~B() {
        cout << " Member function of B" << endl;
      }
};
class C : public A {
   public:
```

```
    C() {
        cout << " Member function of C" << endl;
    }
    ~C() {
        cout << " Member function of C" << endl;
    }
};
int main() {
   B b;
   C c;
}
```

OUTPUT:

```
/Users/yaswanthkakarla/CLionProjects/untitled5/cmake-build-debug/untitled5
Constructor of A
Constructor of B
Constructor of A
Constructor of C
Destructor of C
Destructor of A
Destructor of B
Destructor of A

Process finished with exit code 0
```

v) Hybrid Inheritance:

Program:

```
#include<iostream>
using namespace std;
class Student
{
	protected:
		string name;
		int rollno;
		void getst()
		{
			cout<<"Enter name & rollno: ";
			cin>>name>>rollno;

		}
		void showst()
		{
			cout<<"Name:"<<name<<endl;
			cout<<"RollNo:"<<rollno<<endl;
		}
};
class marks : public Student
{
	protected:
		int m1,m2,m3,m4,m5;
		void getm()
		{
```

```
                getst();
                cout<<"Enter m1,m2,m3,m4,m5 marks:";
                cin>>m1>>m2>>m3>>m4>>m5;
        }
};
class SABL
{
    protected:
        int a1,a2,a3,a4,a5;
        void geta()
        {
            cout<<"Enter a1,a2,a3,a4,a5 points: ";
            cin>>a1>>a2>>a3>>a4>>a5;
        }
};
class percentage: public marks, public SABL
{
    protected:
        float per;
    public:
        void showp()
        {
            getm();
            geta();
            per=(float)(m1+m2+m3+m4+m5+a1+a2+a3+a4+a5)/10;
            showst();
            cout<<"Percentage="<<per;
```

```
        }
};
int main()
{
    percentage p;
    p.showp();

    return 0;
}
```

OUTPUT:

```
/Users/yaswanthkakarla/CLionProjects/untitled5/cmake-build-debug/untitled5
Enter name & rollno: k.yaswanth
171
Enter m1, m2, m3, m4, m5 marks: 50 60 80 90 70
Enter a1, a2, a3, a4, a5 points: 95 90 85 80 75
Name: k.yaswanth
RollNo: 171
Percentage = 77.5%

Process finished with exit code 0
```

EXP.NO: 3.4

AIM: Write a cpp program to display the order of execution of constructor and destructor in inheritance.

Program:

```
#include<iostream>
using namespace std;
class A {
   public:
A() {
        cout << " Member function of A" << endl;
      }
      ~A() {
        cout << " Member function of A" << endl;
      }
};
class B : public A {
   public:
B() {
        cout << " Member function of B" << endl;
      }
      ~B() {
        cout << " Member function of B" << endl;
      }
};
class C : public B {
```

```
    public:
C() {
        cout << " Member function of C" << endl;
    }
    ~C() {
        cout << " Member function of C" << endl;
    }
};
int main() {
    C c;

    return 0;
}
```

OUTPUT:

```
/Users/yaswanthkakarla/CLionProjects/untitled5/cmake-build-debug/untitled5
Constructor of A
Constructor of B
Constructor of C
Destructor of C
Destructor of B
Destructor of A

Process finished with exit code 0
```

EXP.NO: 3.6

AIM: Write a cpp program to illustrate virtual function.

Program:

```
#include<iostream>
using namespace std;
class Base
{       public:
                virtual void show()
                {       cout<<"show() of base class"<<endl;
                }
};
class Derived:public Base
{       public:
                void show()
                {       cout<<"show() of derived class"<<endl;
                }
};
int main()
{
        Base b, *bptr;

        Derived d;
        bptr=&b;
        bptr->show();
        bptr=&d;
        bptr->show();
        return 0;
}
```

OUTPUT:

```
/Users/yaswanthkakarla/CLionProjects/untitled5/cmake-build-debug/untitled5
show() of base class
show() of derived class

Process finished with exit code 0
```

EXP.NO: 3.7

AIM: Write a cpp program to implement pure virtual function

Program:

```
#include<iostream>
using namespace std;
class Base
{       public:
            virtual void show()=0;
};
class Derived:public Base
{       public:
            void show()
            {   cout<<"example of virtual()";
            }
};
int main()
{       Derived d;
        Base *bptr;
        bptr=&d;

        bptr->show();
        return 0;
}
```

Output:

```
/Users/yaswanthkakarla/CLionProjects/untitled5/cmake-build-debug/untitled5
example of virtual()

Process finished with exit code 0
```

AIM: Write a cpp program to calculate the area of different shapes by using abstract class

Program:

```
#include<iostream>
using namespace std;
class Shapes
{       public:
        virtual void area()=0;
};
class Rectangle:public Shapes
{     public:
        int l,b;
 Rectangle()
        {
            l=20;
            b=40;
        }
  void area()
        {
            cout<<"area of rectangle: "<<l*b<<endl;
        }
};
class circle:public Shapes
{
    public:
        int r;
        circle()
        {
            r=8;
```

```
        }
        void area()
        {
            cout<<"area of circle: "<<3.14*r*r<<endl;
        }
};
int main()
{
    Shapes *ptr;
    circle c;
    Rectangle r;
    ptr = &c;
    ptr->area();
    ptr = &r;
    ptr->area();
  return 0;
}
```

Output:

```
/Users/yaswanthkakarla/CLionProjects/untitled5/cmake-build-debug/untitled5
Area of circle: 200.96
Area of rectangle: 800

Process finished with exit code 0
```

EXP.NO: 3.5

AIM: Write a cpp program to illustrate the use of virtual base class

Program:

```
#include<iostream>
using namespace std;
class A
{
        public:
            void showA()
            {
                cout<<"show of class A"<<endl;
            }
};
class B:virtual public A
{
        public:
            void showB()
            {
                cout<<"show of class B"<<endl;
            }
};
class C:virtual public A
{
        public:
            void showC()
            {
                cout<<"show of class C"<<endl;
            }
};
```

```
class D: public B,public C
{
        public:
            void showD()
            {
                cout<<"show of class D"<<endl;
            }
};
int main()
{
        D d;
        d.showD();
        d.showB();
        d.showC();
        d.showA();
        return 0;
}
```

Output:

```
/Users/yaswanthkakarla/CLionProjects/untitled5/cmake-build-debug/untitled5
show of class D
show of class B
show of class C
show of class A

Process finished with exit code 0
```

AIM: Write a cpp program to illustrate pointer to a class

Program:

```
#include<iostream>
using namespace std;
class Base
{
        public:
            void show()
            {
                cout << "Show() of base class" << endl;
            }
};
class Derived: public Base
{
        public:
            void print()
            {
                cout << "print() of derived class" << endl;
            }
};
int main()
{
        Derived d, *dptr;

        dptr = &d;
        dptr->show();
        dptr->print();
        return 0;
}
```

Output:

```
/Users/yaswanthkakarla/CLionProjects/untitled5/cmake-build-debug/untitled5
show() of base class
print() of derived class

Process finished with exit code 0
```

AIM: Write a cpp program to illustrate the use this pointer

Program:

```
#include<iostream>
using namespace std;
class Rectangle
{
        private:
                int length, breadth;

                public:
                void input(int length, int breadth)
                {
                        this->length = length;

                        this->breadth = breadth;
                }
                void area()
                {
                                cout << "Area=" << length * breadth;
                }
};
int main()
{
        Rectangle r;

        r.input(10,30);

        r.area();
}
```

Output:

```
/Users/yaswanthkakarla/CLionProjects/untitled5/cmake-build-debug/untitled5
Area = 300

Process finished with exit code 0
```

AIM: Write a cpp program to illustrate the use of object as a class member

Program:

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

class Birth {
public:
   int dd, mm, YY;

   Birth() {
      dd = 0;
      mm = 0;
      YY = 0;
   }
   void show() {
      cout << "Enter day, month, year: ";
      cin >> dd >> mm >> YY;
      cout << "Date of Birth = " << dd << "-" << mm << "-" << YY << endl;
   }
};

class Student {
public:
   char name[30];
   Birth dob;

   char gender;
```

```
    void print() {
        cout << "Enter name and gender: ";
        cin >> name >> gender;
        cout << "Name = " << name << endl;
        cout << "Gender = " << gender << endl;
        dob.show();
    }
};

int main() {
    Student s;
    s.print();
}
```

Output:

```
/Users/yaswanthkakarla/CLionProjects/untitled5/cmake-build-debug/untitled5
Enter name and gender: k.yaswanth M
Name = k.yaswanth
Gender = M
Enter day, month, year: 29 09 2006
Date of Birth = 29-9-2006

Process finished with exit code 0
```