EXPERIMENT 6

① Write a program to implement single inheritance.

② Write a program to implement multiple inheritance
Problem: Create two base classes Academic & Sports.
- Academic contains marks of a Student.
- Sports contains marks of a Student.

Create a derived class Result that inherits from both Academic & Sports. Write a function to calculate the total score and display details.

③ Create a class Vehicle with attributes like brand and model. Derive a class Car from Vehicle which adds an attribute type (eg. sedan, suv etc) Further derive a class Electric car which adds battery capacity.

④ Create a base class Employee with attributes empID and name. Derive two classes Manager and Developer from Employee.
Manager has an attribute department and developer has an attribute programming language.

⑤ Combine multilevel and multiple inheritance.
Create a base class Person with attributes name and age. Derive a class Student from Person. Create two classes Sports & Academics. Derive class Result from Student & Sports. Write functions to calculate & display total marks & sports score along with Student details.

①
```
#include <iostream>
#include <string>
using namespace std;
class Person
 {
 protected :
string name;
int age;
};
class Student : protected Person
 {
 public:
int roll;
void accept()
 {
cout <<" Enter name of the person:";
cin >> name;
cout <<" Enter age of the person:";
cin>> age;
cout <<" Enter roll number of the person:";
 cin>>roll;
}
void display()
 {
 cout <<endl;
 cout << "Name : " << name <<endl;
 cout << "Roll Number : " << roll. <<endl;
 cout << " Age :" << age <<endl;
}
};
int main()
```

```
{
Student s;
s.accept();
s.display();
return 0;
}
```

Output - Enter name of the person: Anushka
Enter age of the person: 17
Enter roll number of the person: 59

Name: Anushka
Roll Number: 59
Age: 17

(2)
```
#include <iostream>
#include <string>
using namespace std;
class Academic
{
protected:
int m;
};
class Sports
{
protected:
int score;
};
class Result : protected Academic, protected Sports
{
private:
```

```
int t ;
public :
void accept ()
 {
cout << " Enter academic marks : " ;
cin >> m ;
cout << " Enter sports marks : ;
cin >> score ;
}
 void calculate ()
  {
  t = m + score ;
  cout << " Total score is : " << t ;
 }
};
int main ()
 {
  Result r ;
  r.accept() ;
  r. calculate () ;
return 0 ;
 }
```

Output - Enter Academic marks : 90
Enter Sports marks : 96
Total score is : 186

(3) • Multilevel Inheritance

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

```
class Vehicle
{
public:
string brand;
string model;
};
class Car : public Vehicle
  {
  protected:
   string type;
};
class ElectricCar : protected Car
 {
 private:
 int capacity;
  public:
   void accept()
    {
    cout<<"Enter car brand:";
    cin>>brand;
   cout<<"Enter car model:";
    cin>>model;
   cout<<"Enter car type:";
    cin>>type;
  cout<<"Enter battery capacity:";
    cin>>capacity;
 }
 };
int main()
{
ElectricCar e;
```

```
e.accept();
e.display();
return 0;
}
```

Output - Enter car brand : tata
Enter car model : nexon
Enter car type : SUV
Enter battery capacity : 2000
Brand : tata
Model : nexon
Type : SUV
Battery : 2000

(4) · Herarchial inheritance

```
#include <iostream>
#include <string>
using namespace std;
class Employee
{
 protected:
 string name;
  int empid;
};
class Manager : public Employee
{
 private:
 String dept;
public:
 void accept()
 {
```

```
cout << " Enter department name : " ;
cin >> dept;
}
void display()
{
cout << " Department : " << dept << endl;
}
};
class Developer : protected Employee
{
private:
string pl;
public:
void accept()
{
cout << " Enter programming language : ";
cin >> pl;
}
void display()
{
cout << " Programming language : " << pl << endl;
}
};
int main()
{
Manager m;
m.accept();
m.display();
Developer d;
d.accept();
d.display();
return 0; }
```

Output – Enter department name : IT
Department : IT
Enter programming language : C++
Programming language : C++

(5)
```
#include<iostream>
#include<string>
using namespace std;
class Person
{
 protected :
  string name;
  int age;
 public :
  void accept()
  {
   cout<<"Enter name :";
   cin>>name;
   cout<<"Enter age :";
   cin>>age;
  }
  void display()
   {
   cout<<"Name :"<<name<<endl;
   cout<<"Age:"<<age<<endl;
  }
};
class Student : public person
 {
 protected :
   int sid;
```

```
public:
void accept()
{
cout << "Enter Student ID : ";
cin >> sid;
}
void display()
{
cout << "Student ID : " << sid << endl;
}
};
class Sports
{
 protected:
 int sscore;
public:
void accept()
{
cout << "Enter sports score :";
 cin >> sscore ;
}
void display()
{
cout << "Sports score : " << sscore << endl;
}
};
class Academies
 {
protected:
 int marks;
public:
```

```
void accept()
{
cout << "Enter academics marks: ";
cin >> marks;
}
void display()
{
cout << "Academics marks:" marks<<endl;
}
};
class Result : public Student, public Sports
{
private:
 Academics academic;
public:
 void display()
 {
cout << "Total score:" << (marks + score)
                    << endl;
 }
};
int main()
{
Person p;
p.accept();
p.display();
Student s;
s.accept();
 s.display();
Academics a;
a.accept();
```

```
a.display();
Result r;
r.accept();
r.display();
}
```

Output – Enter name : Anushka
Enter age : 16
Enter student ID : 1740
Enter sports score : 85
Enter academic marks : 90

Name : Anushka
Age : 16
Student ID : 1740
Sport score : 85 Academic score : 90
Total score : 175

x ——— x ——— x

6110