Object Oriented Programming

Assignment # 3

Name: Muhammad Suhaib Salman

Roll#: 200768

Class: BEEE-3A

Source Code

```
#include <iostream>

using namespace std;


class Vehicle

{

   protected:

   string cost,number_people;

   string colour,dimension;

   void input()

   {

      cin.ignore();

      cout<<"\n\nEnter cost: ";

      getline(cin,cost);

      cout<<"\nEnter number of  people that can travel in the vehicle: ";

      getline(cin,number_people);

      cout<<"\nEnter dimension: ";

      getline(cin,dimension);

      cout<<"\nEnter colour: ";

      getline(cin,colour);

   }

   void display()

   {

      cout<<"\nThe cost is "<<cost<<endl;
```

```
    cout<<"\nThe number of people that can travel are "<<number_people<<endl;
    cout<<"\nThe dimension is "<<dimension<<endl;
    cout<<"\nThe color is "<<colour<<endl;
  }
};
//////////////////////////////////////////////////////////////////////////
class Car : public Vehicle
{
  protected:
  string transmission_type, back_cam_type;
  void input()
  {
    Vehicle::input();
    cout<<"\nEnter transmission type of the vehicle: ";
    getline(cin,transmission_type);
    cout<<"\nEnter the type of back camera: ";
    getline(cin,back_cam_type);
  }
  void display()
  {
    Vehicle::display();
    cout<<"\nTransmission type of the vehicle: ";
    cout<<"\nThe type of back camera: "<<endl;
  }
```

```
};


class Van : public  Car
{
   string seats,wheeler,van_type;
   public:
   void input()
   {
      Car::input();
      cout<<"\nIs your Van a 4 wheeler or 2 wheeler: ";
      getline(cin,wheeler);
      cout<<"\nEnter number of seats: ";
      getline(cin,seats);
      cout<<"\nEnter the type of van: ";
      getline(cin,van_type);
   }
   void display()
   {
      Car::display();
      cout<<"\nThis van is a "<<wheeler<<endl;
      cout<<"\nThe number of seats in this car are "<<seats<<endl;
      cout<<"\nThe type of the van is "<<van_type<<endl;
   }
};
```

```
class Truck : public Car
{
   string load,truck_type;
   public:
   void input()
   {
      Car::input();
      cout<<"\nEnter maximum load capacity of the truck in kilos: ";
      getline(cin,load);
      cout<<"\nEnter the truck type: ";
      getline(cin,truck_type);
   }
   void display()
   {
      cout<<"\nThe details of the truck are: "<<endl;
      Car::display();
      cout<<"\nThe maximum capacity of the truck is "<<load<<endl;
      cout<<"The type of truck is "<<truck_type<<endl;
   }
};
//////////////////////////////////////////////////////////////////////////

class Boat: public Vehicle
{
```

```
    protected:
    string beam,air_draft,complement;
    void input()
    {
       Vehicle::input();
       cout<<"\nEnter beam: ";
       getline(cin,beam);
       cout<<"\nEnter air draft: ";
       getline(cin,air_draft);
       cout<<"\nEnter the maximum number of people that are required to operate the boat: ";
       getline(cin,complement);
    }
    /*
    Beam – The width of the widest point of the boat
    Draft – The distance between the keel of the boat and the waterline; indicates the
minimum depth of water the vessel needs to float
    Air draft – The distance between the ship's waterline and the highest point of the boat;
indicates the distance the vessel can safely pass under
    Complement – The full number of people necessary to operate a ship, not counting any
passengers
    */
    void display()
    {
       Vehicle::display();
       cout<<"\nThe beam of the boat is "<<beam<<endl;
```

```
        cout<<"The air draft of the boat is "<<air_draft<<endl;
        cout<<"The complement of the boat is "<<complement<<endl;


    }


};
class Sailing_boat: public Boat
{
    string sailing_boat_type;
    public:
    void input()
    {
        Boat::input();
        cout<<"\nPlease enter the type of sailing boat:";
        getline(cin,sailing_boat_type);


    }
    void display()
    {
        Boat::display();
        cout<<"\nThe type of sailing boat is "<<sailing_boat_type<<endl;
    }
};
class Yatchet: public Boat
```

```
{
   string turbo_type,racing;
   public:
   void input()
   {
      Boat::input();
      cout<<"\nPlease enter the turbo type of the motor: ";
      getline(cin,turbo_type);
      cout<<"\nIs your yachet a racing yachet? ";
      getline(cin,racing);



   }
   void display()
   {
      Boat::display();
      cout<<"\nTurbo type: "<<turbo_type<<endl<<"Racing yachet: "<<racing<<endl;
   }
};
class Aircraft : public Vehicle
{
   string max_altitude,fuel_capacity,autopilot;
   public:
   void input()
```

```
    {
        Vehicle::input();
        cout<<"\nPlease enter the maximum altitude that can be achieved by the aircraft: ";
        getline(cin,max_altitude);
        cout<<"\nPlease enter the fuel capcity: ";
        getline(cin,fuel_capacity);
        cout<<"\nDoes your aircraft have autopilot? ";
        getline(cin,autopilot);

    }
    void display()
    {
        Vehicle::display();
        cout<<"\nMax limit of altitude: "<<max_altitude<<endl<<"\nAutopilot support:
"<<autopilot<<endl<<"Fuel capacity: "<<fuel_capacity<<endl;
    }
  };
  class helicopter: public Aircraft
  {
    string tail_rotor_type,rotor_head_type;
    public:
    void input()
    {
        Aircraft::input();
```

```
        cout<<"\nPlease enter the tail rotor type: ";
        getline(cin,tail_rotor_type);
        cout<<"\nPlease enter the rotor head type: ";
        getline(cin,rotor_head_type);

    }
    void display()
    {
        Aircraft::display();
        cout<<"Tail Rotor Type: "<<tail_rotor_type<<endl;
        cout<<"Rotor Head Type: "<<rotor_head_type<<endl;
    }

};

class UFO : public Aircraft
{
    string teleportation_ability,solar_gun,shrink_beam;
    public:
    void input()
    {
        Aircraft::input();
        cout<<"\nDoes your UFO have teleportation ability: ";
        getline(cin,teleportation_ability);
```

```
        cout<<"\nDoes your UFO have solar gun: ";
        getline(cin,solar_gun);
        cout<<"\nDoes your UFO have shrink beam: ";
        getline(cin,shrink_beam);

    }

    void display()
    {
        Aircraft::display();
        cout<<"Solar Gun: "<<solar_gun<<endl<<"Shrink Beam:
"<<shrink_beam<<endl<<"Teleportation Ability: "<<teleportation_ability<<endl;
    }
};

int main()
{
    cout<<"\nWhat type of vehicle do you want?"<<endl;
    cout<<"Press 1 for Van"<<endl<<"Press 2 for Truck"<<endl;
    cout<<"Press 3 for Sailing Boat"<<endl<<"Press 4 Yachet"<<endl;
    cout<<"Press 5 for helicopter"<<endl<<"Press 6 for UFO"<<endl;
    int x=0;
    cin>>x;
```

```
if (x==1)
{
   Van v;
   v.input();
   v.display();
}
else if( x==2)
{
   Truck t;
   t.input();
   t.display();
}
else if (x==3)
{
   Sailing_boat s;
   s.input();
   s.display();
}
else if (x==4)
{
   Yatchet y;
   y.input();
   y.display();
}
```

```
    else if (x==5)
    {
        helicopter h;
        h.input();
        h.display();
    }
    else if (x==6)
    {
        UFO u;
        u.input();
        u.display();
    }
    else
    {
        cout<<"\nPlease enter a correct value!"<<endl;
    }
}
```

OUTPUT

```
suhaib200110@suhaib200110-Latitude-E7270:~/OOP_Semester_3/OOP Theory$ ./a.out

What type of vehicle do you want?
Press 1 for Van
Press 2 for Truck
Press 3 for Sailing Boat
Press 4 Yachet
Press 5 for helicopter
Press 6 for UFO
6
```

```
Enter cost: 200,000,000 Dollars

Enter number of  people that can travel in the vehicle: 10,000

Enter dimension: 777x555x50 feet

Enter colour: black

Please enter the maximum altitude that can be achieved by the aircraft: infinity

Please enter the fuel capcity: 500,000 gallons

Does your aircraft have autopilot? yes

Does your UFO have teleportation ability: yes

Does your UFO have solar gun: yes

Does your UFO have shrink beam: yes

The cost is 200,000,000 Dollars

The number of people that can travel are 10,000

The dimension is 777x555x50 feet

The color is black

Max limit of altitude: infinity

Autopilot support: yes
Fuel capacity: 500,000 gallons
Solar Gun: yes
Shrink Beam: yes
Teleportation Ability: yes
```

Working:

- Firstly I created a parent class Vehicle that contained the common features that were required in its child classes. There was an input function and a display function in the vehicle class.
- Then I derived three classes from the vehicle class. In these classes I described some features of that class and then asked the user to enter specify these features and this class also contained a display function. Thus, all the derived classes had two functions that were input and display function.
- After that I further derived two more classes from each class that was derived from the vehicle class.
- These classes also had the input and display function.
- I used public inheritance in all the derived classes.
- In derived class, in each function I called a function from its parent class for example in input function of UFO class I firstly called the input function of Aircraft class and in Aircraft's class input function I called the input function of the Vehicle class.