OOP Task #2

Q1:

Create class **Vehicle** which contains:

- Model (string)
- Reg_number (string)
- Speed (int)
- Fule_capacity (double)
- Fule_consumption (double)
- Default & parameterized constructors
- Setters & getters
- Double fuelNeeded(int dis) => method that will take distance then calculate the amount of fuel that will be needed for that distance as follow : (fuelNeeded = fuelConsumption * distance).
- double distanceCovered(int hours) => method that will take time (in hours) as an argument and calculate the distance for the given number of hours as follow : (distance = vehicleSpeed * hours)
- display method that will display vehicle information .

Create class **Truck** which inherits from **Vehicle** class and contains following attributes :

- cargo_weight_limit (int)
- Default & parameterized constructors
- A display() method which will call parent display() to print Truck information , and it will print cago_weight_limit with other Truck information's .

Create class **Bus** which inherits from **Vehicle** class and contains following attributes

- Number_of_passengers (int)
- Default & parameterized constructors
- A display() method which will call parent display() to print Bus information , and it will print Number_of_passengers with other Bus information's .

In main :

- Create 3 objects – object of each class , then print each object information .

Q2 :

Create class Movable (abstracted) which contains only the following:

- moveUp() => pure virtual method to achieve abstraction
- moveDown() => pure virtual method to achieve abstraction
- moveLeft() => pure virtual method to achieve abstraction
- moveRight() => pure virtual method to achieve abstraction

Create class MovablePoint which inherits from Movable class , and contain following attributes :

- int x
- int y
- int xSpeed
- int ySpeed
- Default & parameterized constructors
- Implement the above methods as this :
    - moveUp() => increase the value of y by ySpeed
    - moveDown() => decrease the value of y by ySpeed
    - moveLeft() => decrease the value of x by xSpeed
    - moveRight() => increase the value of x by xSpeed

your main could be as this :

```
int main()
{                   //x  y  x_s y_s
      MovablePoint m(5, 5, 2, 3);
      m.moveUp(); // x = 5, y = 8
      m.moveLeft(); // x = 3, y = 8
      m.display_info();
}
```

Q3 : What is the difference between interface vs. abstract class .