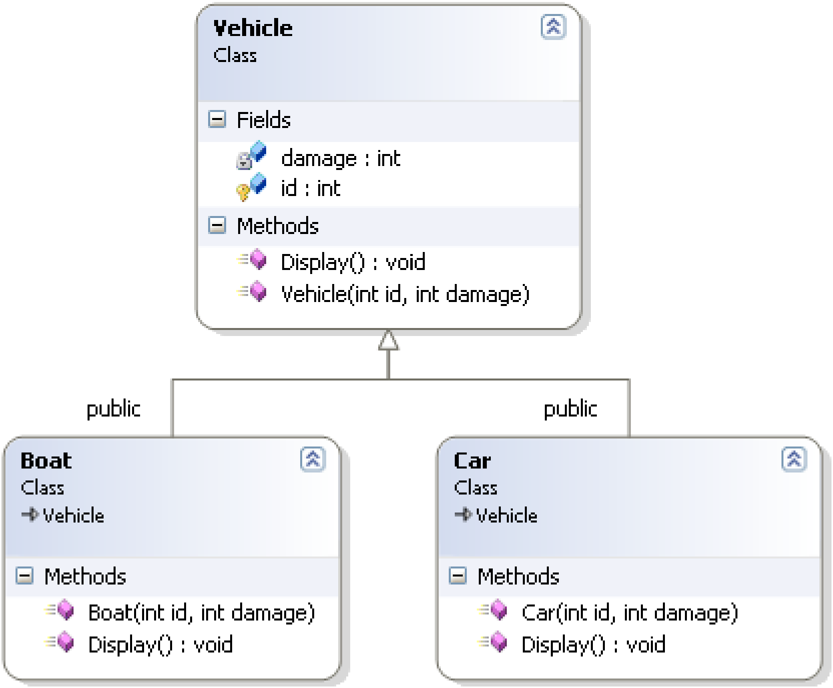
C++ Week 5 Polymorphism

Create a new C++ console application

Exercise 1

Create the class hierarchy illustrated. Ensure you include all the constructors. When creating a sub class the information passed to their constructors should be passed onto the Parent class's constructor where appropriate. The display member function should output the name of the class and the objects id to the console.



Within main create a dynamic instance of each class, passing the following data.

Identifier Name	ID	Damage
V1	1	30
B1	2	50
C1	3	40

Now invoke the display member function on each and run the program.

Exercise 2

Modify the Display member functions of Boat and Car so that they invoke their parent's Display member function. Test it by running the program.

C++ Week 5 Polymorphism

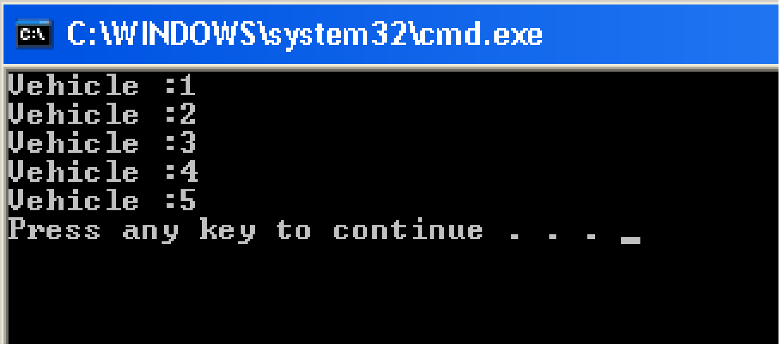
Exercise 3

Comment out the code within main and the statements that invoked the Vehicle's Display member function from within its child classes (exercise 2).
Now create a **static array of pointers**, that point to objects instantiated from the Vehicle class. The array should contain 5 elements each one being assigned one of the objects below, in the order stated

Class Name	ID	Damage
Car	1	35
Boat	2	55
Vehicle	3	30
Car	4	25
Boat	5	65

Within a loop invoke each of the array element's Display member functions.

The output should be as illustrated below. Why is it always outputting vehicle and not Boat and Car?

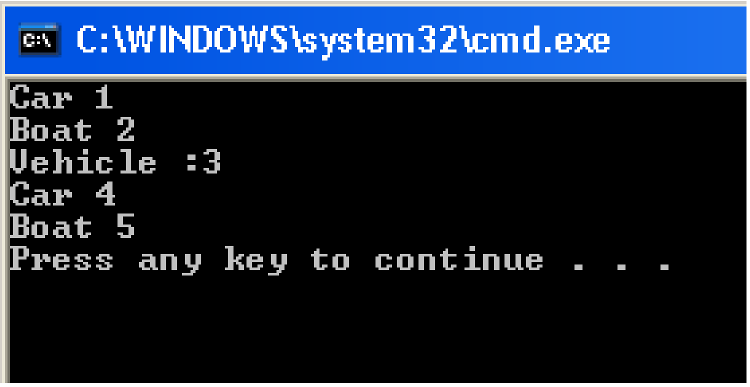


```
Vehicle :1
Vehicle :2
Vehicle :3
Vehicle :4
Vehicle :5
Press any key to continue . . . _
```

C++ Week 5 Polymorphism

Exercise 4

Modify the code so that it exhibits polymorphic behaviour. Your output should resemble that below.

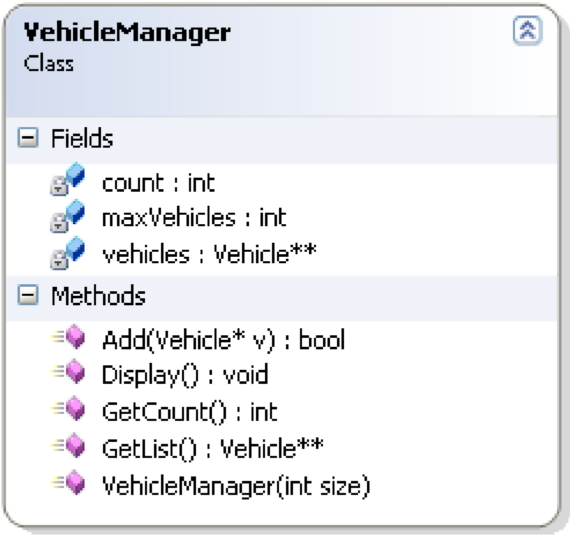


```
Car 1
Boat 2
Vehicle :3
Car 4
Boat 5
Press any key to continue . . .
```

We shall now define a new class that will be responsible for managing the vehicle objects. Within this class, will be an array that stores the vehicles.

Exercise 5

Comment out the code within main and define the VehicleManager class below. See the textual description of each member.



VehicleManager
Class

Fields

- count : int
- maxVehicles : int
- vehicles : Vehicle**

Methods

- Add(Vehicle* v) : bool
- Display() : void
- GetCount() : int
- GetList() : Vehicle**
- VehicleManager(int size)

C++ Week 5 Polymorphism

Exercise 5

- Count- is the number of vehicles within the vehicles array.
- maxVehicles - is the size of the vehicles array which is passed to the constructor in the size parameter.
- Vehicles – a dynamic array of pointers.
- Add – Adds the Vehicle pointed to by v to the array. If successful it returns true otherwise if the array is full it returns false.
- Display – iterates through the vehicles invoking their display member functions.
- GetCount – returns the value in the count data member
- GetList returns the vehicle array
- The constructor creates the array and sets count to zero

Exercise 6

Within main create an instance of VehicleManager that can store 10 vehicles within its array. Add 5 objects to it (use the same types and arguments as in Exercise 3). Finally invoke the manager's Display member function.

Exercise 7

Above main() implement a standalone (friend) function (named DamagedVehicles) that will return an array containing all the vehicles that have sustained 35 or more units of damage. The declaration for the function is

```
Vehicle** DamagedVehicles(Vehicle **vehicles,int *size)
```

From within main invoke the function passing it as an argument the array of vehicles held within the VehicleManager object. Then iterate through the array returned by DamagedVehicles invoking each vehicle's display member function.

Exercise 8

Create a member function within VehicleManager that does a similar task to exercise 7, except it simply invokes the damaged vehicles display member functions rather than adding them to a list.