

# CST8132 Object Oriented Programming

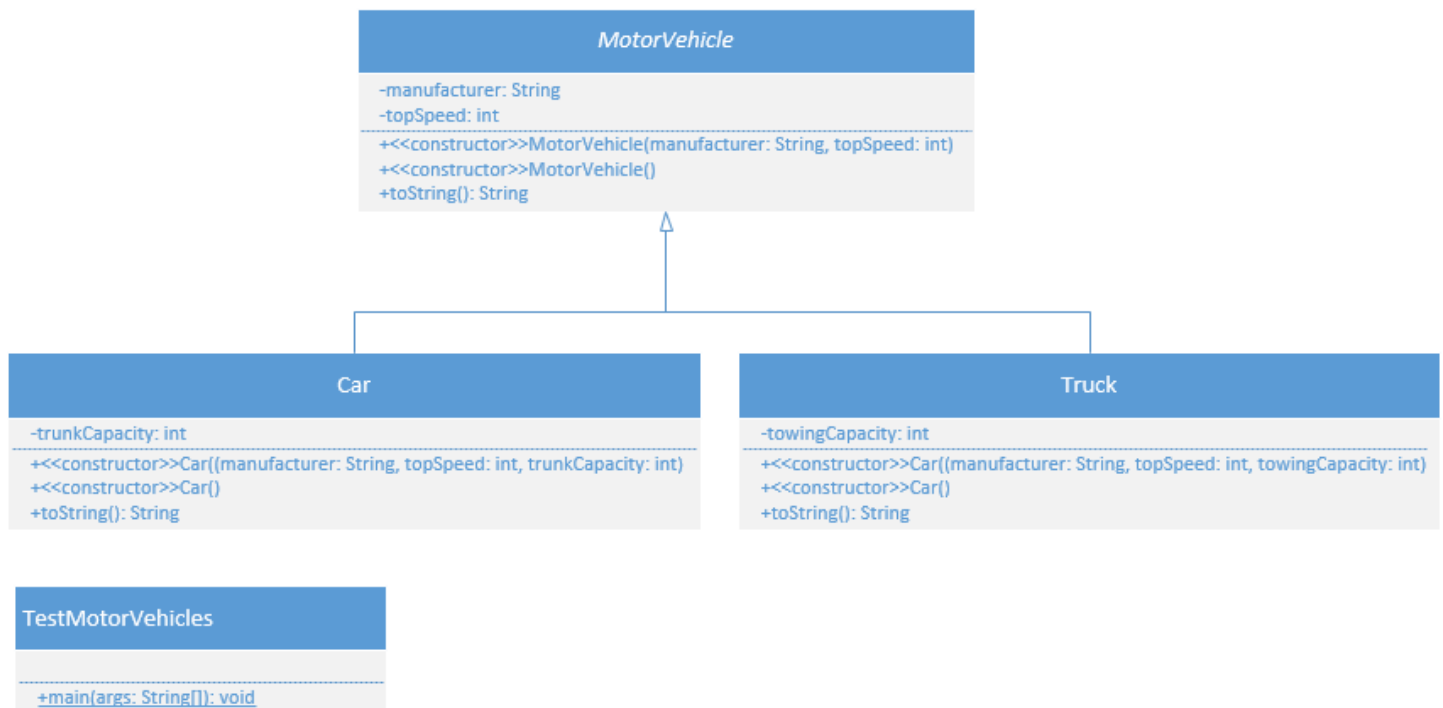
Exercise 4 – Overloaded constructors, inheritance, overriding, use of super() and super.

**Due in the lab period of week 6**

## Steps:

For testing purposes use the TestMotorVehicles class provided.

Use the UML diagrams and the notes below to create a small demonstration program.



## Class MotorVehicle:

- Overloaded constructor:
  - set values into fields from parameters
  - `System.out.println("Returning from MotorVehicle constructor");`
- No parameters constructor:
  - Use a chained constructor call with `this()`
    - Set manufacturer to "UnknownMotorVehicle", set top speed to zero
  - `System.out.println("Returning from default MotorVehicle constructor");`
- `toString`
  - `@Override`
  - Use a `StringBuilder` to append the String returned from the superclasses `toString`
  - Then append the manufacturer and top speed
  - Return the string

## Class Car

- extends MotorVehicle
- one field **trunkCapacity**
- Overloaded constructor:
  - Explicitly call super(String, int) passing values into super class
  - set value into field from last parameter for trunkCapacity
  - System.out.println("Returning from Car constructor");
- No parameters constructor:
  - Use a chained constructor call with this()
    - Set manufacturer to "UnknownCar", set top speed to 100, trunk capacity to 1
  - System.out.println("Returning from default Car constructor");
- toString
  - @Override
  - Use a StringBuilder to append the String returned from the superclasses toString
  - Then append the trunkCapacity
  - Return the string

## Class Truck

- extends MotorVehicle
- one field **towingCapacity**
- Overloaded constructor:
  - Explicitly call super(String, int) passing values into super class
  - set value into field from last parameter for towingCapacity
  - System.out.println("Returning from Truck constructor");
- No parameters constructor:
  - Use a chained constructor call with this()
    - Set manufacturer to "UnknownTruck", set top speed to 95, towing capacity to 11,000
  - System.out.println("Returning from default Truck constructor");
- toString
  - @Override
  - Use a StringBuilder to append the String returned from the superclasses toString
  - Then append the towingCapacity
  - Return the string

```
public class TestMotorVehicles {
public static void main(String[] args){
    MotorVehicle mv1 = new MotorVehicle(); // OPPORTUNITY FOR BONUS MARKS
    System.out.println(mv1);
    System.out.println();

    MotorVehicle mv2 = new Car();
    System.out.println(mv2);
    System.out.println();

    MotorVehicle mv3 = new Car("SampleData1", 125, 2);
    System.out.println(mv3);
    System.out.println();

    MotorVehicle mv4 = new Truck();
    System.out.println(mv4);
    System.out.println();

    MotorVehicle mv5 = new Truck("SampleData2", 95, 12000);
    System.out.println(mv5);
    System.out.println();
}
}
```

## Grading Guide (Total Score 10):

Correct use of extends?	2
Super class constructors chained with super()	3
toString() overridden, with calls to super.toString()	3
Program produces correct output	2

\*\*\*\*\*

### Opportunity for bonus marks

1 bonus mark: What class would the MotorVehicle class extend? Add appropriate code and demo.

1 bonus mark: Add keyword **abstract** to the class definition for the MotorVehicle class. You should now have errors in your main method. Debug your code and be prepared to answer why.

Maximum available mark for this lab: 12/10

\*\*\*\*\*

## Sample Program Output

Returning from overloaded MotorVehicle constructor

Returning from default MotorVehicle constructor

MotorVehicle@2a139a55 manufacturer: UnknownMotorVehicle topSpeed: 0

Returning from overloaded MotorVehicle constructor

Returning from overloaded Car Constructor

Returning from default Car Constructor

Car@15db9742 manufacturer: UnknownCar topSpeed: 100 trunkCapacity: 1

Returning from overloaded MotorVehicle constructor

Returning from overloaded Car Constructor

Car@6d06d69c manufacturer: SampleData1 topSpeed: 125 trunkCapacity: 2

Returning from overloaded MotorVehicle constructor

Returning from overloaded Truck Constructor

Returning from default Truck Constructor

Truck@7852e922 manufacturer: UnknownTruck topSpeed: 100 towingCapacity: 11000

Returning from overloaded MotorVehicle constructor

Returning from overloaded Truck Constructor

Truck@4e25154f manufacturer: SampleData2 topSpeed: 95 towingCapacity: 12000

## Recommended Readings on UML:

Donald Bell. (2004). UML basics: The class diagram. Retrieved from

<http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/#N100A5>

Kirill Fakhroutdinov. (2014). Class Diagrams. Retrieved from

<http://www.uml-diagrams.org/class-diagrams.html>

Hippias Minor. (2013). stackoverflow: Is there a specific notation for a final static class in UML?

Retrieved from: <http://stackoverflow.com/questions/17165638/is-there-a-specific-notation-for-a-final-static-class-in-uml>