

# COMP2232 - Lab Practical #1

## IT IS IMPORTANT READ THIS ENTIRE DOCUMENT BEFORE YOU BEGIN

*Purpose: to facilitate your familiarity with basic Java language and object-oriented programming concepts.*

### *The Scanner Class*

---

Java includes a **Scanner** class which allows for simple keyboard input. To use this class, you will need to import it. See example below:

```
import java.util.Scanner; //You will need to import the Scanner class

public class MyClass
{
    public static void main(String[] args)
    {
        //Create an object of type Scanner
        Scanner myScanner = new Scanner(System.in);

        /* You can now use methods in the Scanner class via your myScanner object
        to read data from the keyboard. There are various methods for different
        types of data. A few are outlined below.*/

        int intNum = myScanner.nextInt(); //Read an integer value from keyboard
        double dblNum = myScanner.nextDouble(); //Read a double value
        String word = myScanner.next(); //Read a single word from keyboard
        String message = myScanner.nextLine(); //Read an entire line of text
    } // end - main
} //MyClass
```

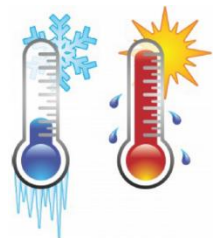
### *A - Temperature Converter*

---

1. Create a **TemperatureConverter** class which will contain:
  - a. a function which accepts a temperature in Celsius and **returns** the temperature in Fahrenheit. Use the formula  $Fahrenheit = ((9/5) \times Celsius) + 32$ .
  - b. a function which accepts a temperature in Fahrenheit and **returns** the temperature in Celsius. Convert the formula above to work out the Celsius value.

*Note: neither of these methods displays to the screen.*

2. Create a **ConversionTester** (driver) class which will:
  - a. Ask the user to enter a temperature and choose whether they want to convert it to Celsius or Fahrenheit.
  - b. Instantiate a *TemperatureConverter* object and use the appropriate method to complete the conversion and return the result. Display the result for the user.



## B – The Vehicle Class

---

1. Create and implement a class called **Vehicle** based on the UML diagram below:

Vehicle
- model: String - engineSize: double - currSpeed: int
+ Vehicle() <<constructor>> + setModel(String s): void + getModel(): String + setEngineSize(double d): void + getEngineSize(): double + accelerate(int speed): void + brake(): void

a. **Vehicle**: constructor which will initialize **model** to “basic”, **engineSize** & **currSpeed** to 0.0 & 0, respectively.

b. **setModel** and **getModel**: mutator & accessor methods (respectively) for the model data member.

c. **setEngineSize** and **getEngineSize**: mutator & accessor methods (respectively) for the engineSize data member. EngineSize must not be a negative value.

d. **accelerate** method will display the speed as it increases (by 5km increments) until it reaches the maximum value (which was passed to the function using the parameter).

e. **brake** method will display the speed as it decreases (by 5km decrements) until it reaches 0. It will then display the message “You have safely come to a stop.”

2. Create a class called **CarTest** which contains a **main** method, implementing the following within it:

a. Ask the user for the model and engine size of a vehicle.

b. Instantiate a **Vehicle** object called **myCar**.

c. Utilize the object’s mutators to store the data in the object.

d. Once done, ask the user for a speed for the vehicle and use the object’s accelerate method to implement the user’s request.

e. Create a loop which will ask the user if they wish to stop. If their response is negative, the car continues at the same speed (which is displays) and the user is again asked if they wish to stop. If they respond positively, then the brake method will be invoked. Note: for user responses, you can choose to accept either Strings (“yes” / “no”), characters (‘y’ / ‘n’) or numbers (0/1). The design choice is yours.

f. Once the vehicle has come to a complete stop, utilize the object’s accessors to display the vehicle’s details (model & engine size) and thank the user for taking the vehicle for a test drive.



<End of Lab>

