Montgomery College, CMSC 203
Worksheet 1

Module 9 Objectives

- JUnit

Concept Questions

1) Which of the following is correct about manual testing?
   A) Since test cases are executed by human resources, it can be very slow and tedious.
   B) Since test cases need to be executed manually, more labor (test programmers) are required
   C) Both of the above.
   D) None of the above.

2) What is unit testing?

3) What is a Unit Test Case?

4) What is JUnit?

5) What is `@Test` and where is it used?

6) What is `@Before` and `@After` and its usage?

7) Which methods cannot be tested by JUnit test class?
   A) public methods
   B) private methods
   C) protected methods
   D) methods with void return type

Programming Questions

1. Given the following `Conversion` class, implement the following:

```
public class Conversion {
	private double temp;  // Temperature

	public Conversion ( double temp)
	{
		this.temp = temp;
	}

	public double tempConversion(String unit)
	{
		if ( unit.equals("F"))
			return (temp -32 ) * (5.0/9);   //Convert to Celcius
		else
			return (temp * (9.0/5) ) + 32;  // Convert to Fahrenheit
	}
	public String toString ()
	{
		return "Temperature conversion program! " + temp;
	}

}
```

And the following test class , implement the specified tests for the given comments.

```
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

class ConversionTest {
	Conversion c1 ,c2;

	@BeforeEach
	void setUp() throws Exception {
		//Create an object c1 of type Conversion with the temp value set to 100

		//Create an object c2 of type Conversion with the temp value set to 32

	}
	@AfterEach
	void tearDown() throws Exception {
		//destroy the reference to c1 and c2

	}
	@Test
	void testtempConversion() {
		// assert if c1.tempConversion("F") is returning 37.7

		// assert if c2.tempConversion("C") is returning 89.6
	}
	@Test
	void testtoString() {
		// use assertTrue method to test if c1.toString method is returning the
correct result

	}
```

```
}
```

2. Following represent a `Car` Class:

```
public class Car {
	private String name;
	private double price;
	private String ownerName;

	public Car(String name, double price, String ownerName) {
		this.name = name;
		this.price = price;
		this.ownerName = ownerName;
	}

	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public double getPrice() {
		return price;
	}
	public void setPrice(double price) {
		this.price = price;
	}
	public String getOwnerName() {
		return ownerName;
	}
	public void setOwnerName(String ownerName) {
		this.ownerName = ownerName;
	}
}
```

Implement the following method in the Car Class that returns the discounted price of a `Car` object based on a given discount rate:

```
public double discountPrice(double rate)
```

Create the following Junit test class:

- In the setup method create two `Car` instances with the following information:
  `name` : BMW
  `price`: 10000
  `owner`: Bill
- Create a test method which will test the equality of all the fields of the `Car` objects.
- Create a test method which will test discountPrice method. For example a call to
- `discountPrice(10)`, assuming the above information for the Car object should return 9000 .
- In the tear down method delete the `Car` objects which you created.