GIFT School of Engineering and Applied Sciences

Spring 2022

CS-240: Object-oriented Programming

Lab-6 Manual

Basics of Classes and Objects

Task #1: Creating Classes and Objects

In this task, you are being asked to write a class and create objects in Java.

NOTE: Write your class and the *main* method in separate files.

Write a class named **Car** that has the following fields:

- **yearModel.** The **yearModel** field is an **int** that holds the car's year model.

- **make.** The **make** field references a **String** object that holds the make of the car.

- **speed.** The **speed** field is an **int** that holds the car's current speed.

In addition, the class should have the following methods:

- **Accessors**. Appropriate accessor methods should get the values stored in an object's **yearModel**, **make**, and **speed** fields.

- **accelerate**. The **accelerate** method should add **5** to the **speed** field each time it is called.

- **brake**. The **brake** method should subtract **5** from the **speed** field each time it is called.

- **display**. The display method should display the state of the object.

Demonstrate the class in a program that creates a **Car** object, and then calls the **accelerate** method five times. After each call to the **accelerate** method, get the current **speed** of the car and display it. Then call the **brake** method five times. After each call to the **brake** method, get the current **speed** of the car and display it.

1. Create a program called **Car.java**, and **RunCar.java** having the **main** method**.**
2. Create two Car objects in RunCar.
3. Check which car has more speed in main method.
4. Check which car is older in main method in main method.
5. Call display method for both objects.
6. Correctly display appropriate messages.

Task #2: Creating Classes and Objects

In this task, you are being asked to write a class and create objects in Java.

NOTE: Write your class and the *main* method in separate files.

Write a class named **RetailItem** that holds data about an item in a retail store. The class should have the following fields:

- **description.** The **description** field references a **String** object that holds a brief description of the item.

- **unitsOnHand.** The **unitsOnHand** field is an **int** variable that holds the number of units currently in inventory.

- **price.** The **price** field is a **double** that holds the item's retail price.

Write appropriate **mutator** methods that store values in these fields, and **accessor** methods that return the values in these fields.

Once you have written the class, write a separate program that creates three **RetailItem** objects and stores and then displays the following data in them:

	Description	Units on Hand	Price
Item #1	Jacket	12	59.95
Item #2	Designer Jeans	40	34.95
Item #3	Shirt	20	24.95

1. Create a program called **RetailItem.java**, and **RunRetailItem.java** having the **main method.**
2. Create two RetailItem Objects in RunRetailItem.
3. Check which retail item is costly main method.
4. Check which retail item has more units in inventory in main method.
5. Correctly display appropriate messages.

Task #3: Creating Classes and Objects

In this task, you are being asked to write a class and create objects in Java.

NOTE: Write your class and the *main* method in separate files.

Write a **Temperature** class that will hold a temperature in Fahrenheit, and provide methods to get the temperature in Fahrenheit, Celsius, and Kelvin. The class should have the following field:

- **ftemp** – A **double** that holds a Fahrenheit temperature.

The class should have the following methods:

- **setFahrenheit** – The **setFahrenheit** method accepts a Fahrenheit temperature (as a **double**) and stores it in the **ftemp** field.

- **getFahrenheit** – Returns the value of the **ftemp** field, as a Fahrenheit temperature (no conversion required).

- **getCelsius** – Returns the value of the **ftemp** field converted to Celsius.

- **getKelvin** – Returns the value of the **ftemp** field converted to Kelvin.

Use the following formula to convert the Fahrenheit temperature to Celsius:

> *Celsius = (5/9) × (Fahrenheit - 32)*

Use the following formula to convert the Fahrenheit temperature to Kelvin:

> *Kelvin = ((5/9) × (Fahrenheit - 32)) + 273*

Demonstrate the **Temperature** class by writing a separate program that asks the user for a Fahrenheit temperature. The program should create an instance of the **Temperature** class, ask the user for an input Fahrenheit temperature, and call the **setFahrenheit** method to set the data member value. The program should then call the object's methods to display the temperature in Celsius and Kelvin.

1. Create a program called **Temperature.java**, and **RunTemperature.java** having the **main** method**.**
2. Create two objects of the Temperature in RunTemperature.
3. Set the temperature in both objects.
4. Check which object has more temperature in terms of celcius.
5. Correctly display appropriate messages.

Task #4: Creating Classes and Objects

In this task, you are being asked to write a class and create objects in Java.

NOTE: Write your class and the *main* method in separate files.

The following table lists the freezing and boiling points of several substances.

Substance	Freezing Point	Boiling Point
Ethyl Alcohol	–173	172
Oxygen	–362	–306
Water	32	212

Design a class called **Substance** that stores a temperature in a **temperature (int)** field and has the appropriate accessor and mutator methods for the field. The class should also have the following methods:

- **isEthylFreezing.** This method should return the **boolean** value **true** if the temperature stored in the **temperature** field is at or below the freezing point of ethyl alcohol. Otherwise, the method should return **false**.

- **isEthylBoiling.** This method should return the **boolean** value **true** if the temperature stored in the **temperature** field is at or above the boiling point of ethyl alcohol. Otherwise, the method should return **false**.

- **isOxygenFreezing.** This method should return the **boolean** value **true** if the temperature stored in the **temperature** field is at or below the freezing point of oxygen. Otherwise, the method should return **false**.

- **isOxygenBoiling.** This method should return the **boolean** value **true** if the temperature stored in the **temperature** field is at or above the boiling point of oxygen. Otherwise, the method should return **false**.

- **isWaterFreezing.** This method should return the **boolean** value **true** if the temperature stored in the **temperature** field is at or below the freezing point of water. Otherwise, the method should return **false**.

- **isWaterBoiling.** This method should return the **boolean** value **true** if the temperature stored in the **temperature** field is at or above the boiling point of water. Otherwise, the method should return **false**.

Write a program that demonstrates the class. The program should ask the user to enter a temperature, and then display a list of the substances that will freeze at that temperature and those that will boil at that temperature. For example, if the temperature is **−20** the class should report that **water** will **freeze,** and **oxygen** will **boil** at that temperature.

1. Create a program called **Substance.java**, and **RunSubstance.java** having the **main** method**.**
2. Correctly display appropriate messages.