CSC72002 Assignment 2

Weight: 40% of your final mark

Due: 12 Feb 2021 11 pm

Specifications

Your task is to complete various exercises in NetBeans, using the Java language, and to submit these via the MySCU link created for this purpose.

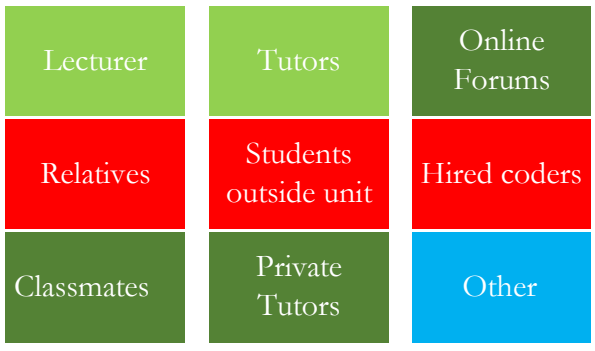
Marking criteria includes:

- Use of correct coding style, including the use of comments;
- Accuracy of coding;
- Use of suitable coding structures;
- Correct submission and naming conventions of assessment items as required.

Getting Help

This assignment is to be completed individually. It is the opportunity to gain an understanding of the concepts of object-oriented programming and coding syntax. It is important that you master these concepts yourself. You are permitted to work from the examples in the study guide or textbook, but you must acknowledge assistance from other textbooks or classmates. In particular, you **must not** use online material or help from others, as this would prevent you from mastering these concepts.

Who can you get help from? Use this diagram to determine from whom you may seek help with your program.



Encouraged

Attribution Required

Ask tutor

Not acceptable

Please Note for all parts of the assignment:

- You **cannot** use the same examples that the Study guide uses. Examples taken from the study guide will result in no marks and may count as plagiarism.

- Marks will be given for high cohesion. This means that you should not have all your code in the start method.

Part 1 –Shapes and Events

Create a new JavaFX project called **usernamePart1** in NetBeans. In my case, the project would be called rghanbarPart1.

Using a GridPane:

- Add at least **4 different** shapes to your app (Every shape must have at least 1 event)
- Demonstrate the use of at least **4 different** Mouse events
- Demonstrate the use of at least **2 different** Key events
- Demonstrate the use of at least 1 inner class
- Demonstrate the use of at least 1 anonymous inner class
- Demonstrate the use of at least 1 lambda expression
- Demonstrate the use of setHgap, setVgap, setHalignment and setValignment
- Ideally you should have a method for each shape.

Part 2 – Images and Animations

Create a new JavaFX project called **usernamePart2** in NetBeans. In my case, the project would be called rghanbarPart2.

- Add an image as the background for your app using an ImageView object
- Demonstrate the use of at least 3 methods of ImageView e.g. setFitHeight (https://docs.oracle.com/javase/8/javafx/api/javafx/scene/image/ImageView.html)
- Add a Polygon with at least 8 sides
- Demonstrate the use of the PathTransition class using the polygon as the path and any shape as the node
- Add a second shape (can be any shape)
- Demonstrate the use of the Timeline class to change the colour of the shape every 1 second indefinitely e.g. red to blue to red to blue to red to blue etc. You can choose what colours you like

Part 3 – UI Controls and Collections

Create a new JavaFX project called **usernamePart3** in NetBeans. In my case the project would be called rghanbarPart3.
Imagine you are writing an app for a coffee shop to allow the staff to enter orders for the customers. When the user is finished the data will be written to an Order object (see attached class at the end of this assignment) and then the Order object will be added to an ArrayList so that it can be stored. The user can then enter data for a new Order and repeat the process.

Your app will need the following UI controls that will be mapped to instance variables in the Order Class provided at the end of this document:

- Text field for the first name
- Text field for the last name
- Slider for the number of coffees
- Radio buttons for at least 4 coffee types (cappuccino, latte etc)
- A combo box for at least 4 sizes (small, medium etc)
- Check boxes for at least 4 coffee extras (sugar, skim milk, cream etc)
- A list view for at least 4 side orders of food (cake, muffin etc)

Please note that if the user enters an order for more than 1 coffee, all of the coffees in an order will be of the same type.

Once a user has entered the data for an Order your app needs to do the following:

- Write the data from the UI to an Order object (use the attached class at the end of this assignment)
- Add the Order object to a ArrayList that will store all Orders entered by the app
- Reset the UI controls so the user can enter a new Order

One way to do this would be to have the following (you do not have to do it this way):

- A "New Order" button that resets the UI controls.
- An "Add Order" button that creates a new instance of the Order, sets the Orders instance variables (via the mutator methods) using the values from the UI controls and then adds the Order to the ArrayList

PLEASE NOTE:

- Marks will be given for cohesion i.e. you should not have all of your code in the start method. Your program should make use of multiple methods, each with a distinct function
- You CANNOT modify the attached Order class

Submission

You should now have 3 Netbeans projects. You must zip the projects into one zip file called username_A2.zip For example, mine would be rghanbar_A2.zip.
Submit this file via the Assignment 2 link on MySCU by the due date. Please leave enough time for it to upload, so do not submit at the last minute!

Order Class

Use the following class for your Order:

```
public class Order
{         // text
field     private
String firstName;
// text field
private String
lastName;
    // slider
    private Integer numberOfCoffees;
    // radio
buttons
private String
coffeeType;
// combo box
private String
coffeeSize;     //
check boxes
    private HashSet<String> extras = new
HashSet();
    // list view
    private HashSet<String> sideOrder =
new HashSet();

    public Order() {
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String
firstName) {         this.firstName =
firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String
lastName) {         this.lastName =
lastName;
    }

    public Integer getNumberOfCoffees()
{
        return numberOfCoffees;
    }

    public void setNumberOfCoffees(Integer
numberOfCoffees) {         this.numberOfCoffees =
numberOfCoffees;
    }
```

```
    public String
getCoffeeType() {
return coffeeType;
    }

    public void setCoffeeType(String
coffeeType) {         this.coffeeType =
coffeeType;
    }

    public String
getCoffeeSize() {
return coffeeSize;
    }

    public void setCoffeeSize(String
coffeeSize) {         this.coffeeSize =
coffeeSize;
    }

    public HashSet<String> getExtras() {
        return extras;
    }

    public void
setExtras(HashSet<String> extras) {
        this.extras = extras;
    }

    public HashSet<String>
getSideOrder() {
        return sideOrder;
    }

    public void
setSideOrder(HashSet<String> sideOrder)
{
        this.sideOrder = sideOrder;
    }
}
```