**T323  
ICT104 Program Design and Development**

**Week 7 Tutorial**

**Topic: Object Oriented Concepts**

**Objective: Students learn about the basics of Objects and Classes using Java**

**Submission: The completed tutorial word file containing student’s answers need to be  
uploaded to Moodle by Sunday 17/12/2023 @ 23.59**

**Part A:** Highlight the correct option.

1. To end an application, pass this as the argument to the JFrame class's setDefaultCloseOperation() method.

A) END\_ON\_CLOSE

B) JFrame.END\_ON\_CLOSE

C) JFrame.EXIT\_ON\_CLOSE

D) JFrame.CLOSE\_NOT\_HIDE

2. In GUI terminology, a container that can be displayed as a window is known as a:

A) message dialog

B) buffer

C) Swing package

D) frame

3. The minimize button, maximize button, and close button on a window are sometimes referred to as:

A) operations buttons

B) sizing buttons

C) decorations

D) display buttons

4. Programs that operate in a GUI environment must be:

A) event driven

B) in color

C) dialog boxes

D) layout managers

5. \_\_\_\_\_\_\_\_ is a library of classes that do not replace \_\_\_\_\_\_\_\_ but provide an improved alternative for creating GUI applications.

A) AWT, Swing

B) Swing, AWT

C) JFC, AWT

D) JFC, Swing

6. To use the Color class, which is used to set the foreground and background of various objects, use the following import statement:

A) import java.swing;

B) import java.awt;

C) import java.awt.\*;

D) import java.awt.event.\*;

7. Which one of the following GUI components is a container?

A) Frame

B) Label

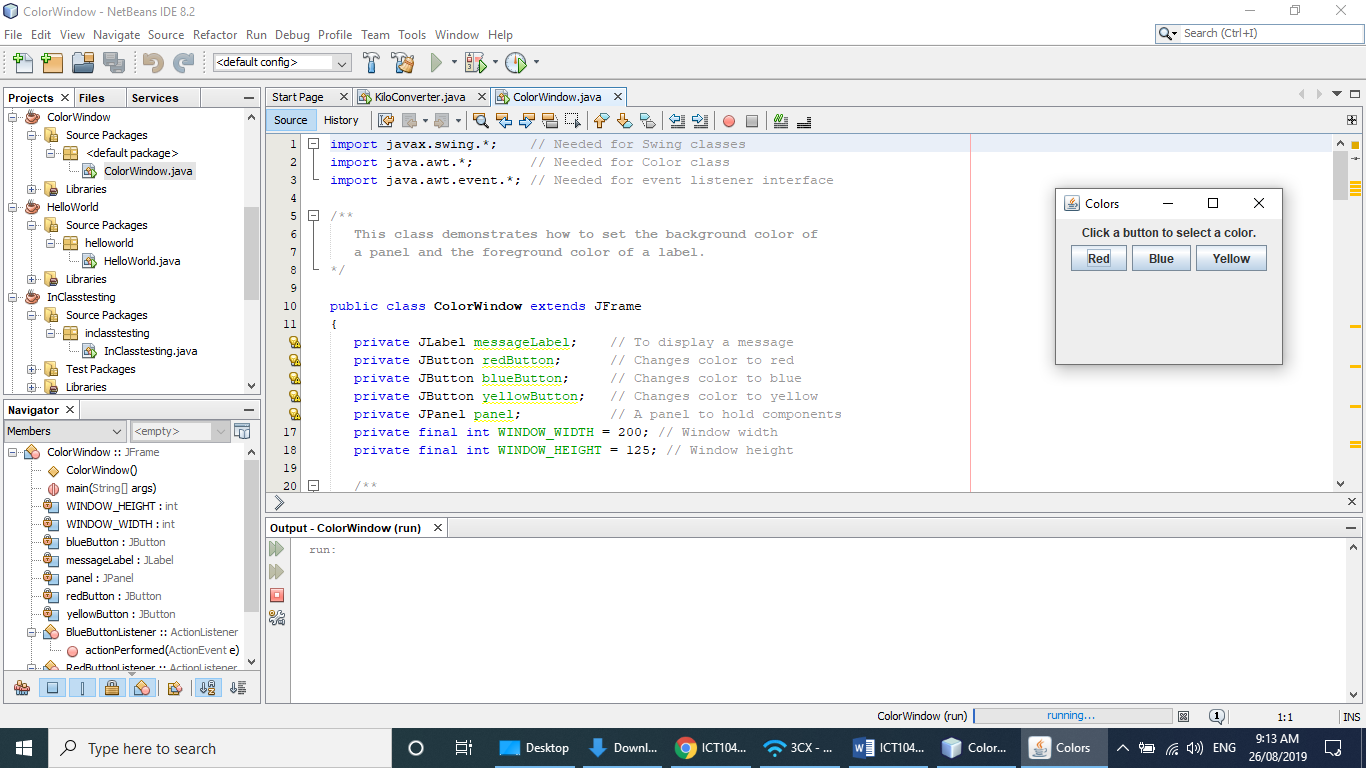
C) Slider

D) Button

**Part B:** State TRUE or FALSE:

1. Some of the common GUI components are buttons, labels, text fields, check boxes, and radio buttons.
2. A GUI program automatically stops executing when the end of the main method is reached.
3. The ActionEvent argument that is passed to an action listener's actionPerformed method is the event object that was generated in response to an event.

**Part C:** Write Java Code to display a window with a label and three buttons as shown in the figure. When the user clicks a button, it changes the background color of the panel that contains the components and the foreground color of the label.



2. Identify the output of the following Java code. Justify the syntax.  
(these can be downloaded from **Moodle** under **Week 7 Lecture 7 T323 Resources**)

**File Name: SimpleWindowDemo.java**

**/\*\***

**This program creates an instance of the**

**SimpleWindow class.**

**\*/**

**public class SimpleWindowDemo**

**{**

**public static void main(String[] args)**

**{**

**SimpleWindow myWindow = new SimpleWindow();**

**}**

**}**

**File Name: SimpleWindow.java**

**import javax.swing.\*; // Needed for Swing classes**

**/\*\***

**This class extends the JFrame class. Its constructor displays**

**a simple window with a title. The application exits when the**

**user clicks the close button.**

**\*/**

**public class SimpleWindow extends JFrame**

**{**

**/\*\***

**Constructor**

**\*/**

**public SimpleWindow()**

**{**

**final int WINDOW\_WIDTH = 350; // Window width in pixels**

**final int WINDOW\_HEIGHT = 250; // Window height in pixels**

**// Set this window's title.**

**setTitle("A Simple Window");**

**// Set the size of this window.**

**setSize(WINDOW\_WIDTH, WINDOW\_HEIGHT);**

**// Specify what happens when the close button is clicked.**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**// Display the window.**

**setVisible(true);**

**}**

**}**

**Part D:** Reflection Exercise

List the name of the topics/concepts you have learnt this week.

**Optional Tasks**

# The TicketSales.Com website sells tickets to a range of entertainment events. The following are some examples:

# •          A single-use ticket to attend a Tourist attraction.

# •          A multi-use ticket to attend a fixed number of entries to tourist attractions.

In this lab we will write a program to simulate the selling and usage of some tickets by the website. Read through all tasks before you begin, to get an idea of what is expected to be completed.

**Task 1: Create Interface definitions**

Create a new package for this week’s activities called, for instance, week7

First, we need to define some interfaces, which we will later implement.

1. Create an Interface named SingleUseTicketing. It will need to define the following methods:

· **use( )** – this method will be called to signify that the ticket is to be used. The method should return true if the ticket is valid for use, and false if it is not valid for use because of having already been used. The effect of the method should be to change the ticket from useable (i.e. valid), to being no longer valid in the future.

2. Create an interface named MultiUseTicketing. It will need to define the following methods:

· **loadUp( amount )** – this method will be called to add a specified amount of uses to the ticket (each is called a use-entitlement). For example if the parameter is 3, this means the ticket can be used 3 times more than it previously could. Thus if the ticket had been fully used-up, it would again be available to be used.

· **useOnce( )** – this method will be called to signify that the ticket is to be used one more time. The method should return true if the ticket is valid for use, and false if it is not valid for use because of having expired, i.e. having been fully used-up. The effect of the method should be to decrease the number of uses that remain for the ticket. For example, if 3 uses had been loaded onto the ticket, then the first call to useOnce will reduce the number to 2.

**Task 2: Create abstract class definition**

Define an abstract class named Ticket. It should provide the following methods, possibly as abstract methods:

• **getPurpose( )** – returns a String describing the purpose for which the ticket may be used, e.g. to attend a sport match, to attend multiple sport matches, to enter the Zoo, etc.

• **getCost( )** – returns an amount signifying the cost of obtaining the ticket.

• **isValid( )** – returns a boolean value indicating if the ticket is valid for use, or not.

**Task 3: Create other class definitions**

1. Create a concrete subclass of Ticket named TouristTicket that implements the

SingleUseTicketing interface.

• One attribute should be the name of the tourist attraction (e.g., Zoo, Water World, etc.)

• Another attribute should be the price of entry to the attraction.

• Ensure that all required methods as specified by the interface or superclass are implemented (with bodies).

2. Create a subclass of Ticket named SportTicket that implements the MultiUseTicketing interface. It will need to ensure that the following information is stored in attributes:

• The cost of the ticket itself, without any use-entitlements.

• The cost for each use-entitlement (each entry/admittance to a match), i.e. to load 4 admittances onto the ticket may cost $17 each when pre-purchased, whereas the base cost of the ticket itself may be just $5 and is charged just once for the whole year.

• How many times it may be used.

 Make sure that you implement the methods required by the MultiUseTicketing interface, so that you check how many uses remain and check the ticket’s validity.

**Task 4: Write a driver class**

You may prefer to do this part gradually as you work through the other tasks.

Write a driver which creates objects of the concrete class types, and calls methods on the objects.

**Task 5: Start preparing for Quiz 2 worth 15% which will be conducted in Week 8.**