NIT2112 Object Oriented Programming
College of Engineering and Science

VICTORIA UNIVERSITY
MELBOURNE AUSTRALIA

‘Unique Six’ Assignment

Due: Week 12 Sunday 27 May 5 pm (Demo in Swotvac week)

No late submission

Marks: The assignment is 15% of final mark for the unit.

You are to work with a partner, 2 students per group. Groups MUST submit for both students, only one copy of the assignment. Submission details are at the end of this document.

The assignment is composed of three parts.

Problem Description

You are to design and implement a class to play a game called *Unique Six.* In this game, the player will roll a die to obtain a number from 1 to 6 and continue rolling until all six unique numbers have been generated. If a number is rolled that has appeared before, then the number is discarded until all the six required numbers are obtained. Your program will check and test if each of these numbers has appeared only once. You may use some of the methods defined in the Die class of Chapter 5 in the textbook to help in your assignment. You may also need minor modifications to the existing class(es) as you progress from Part 1 to Part 3. Remember small victories are keys to success.

Part 1

You need to create three classes, class Die, class Player and class TestSixNumbers.
class Die has the ability to roll the die and produce a new random face value.
class Player creates an instance of class Die and then checks if the numbers 1, 2, 3, 4, 5 and 6, appear only once. It stops rolling the die once a set of six unique numbers from 1 to 6 has been obtained.
class TestSixNumbers creates an instance of class Player and it plays the game.
class TestSixNumbers also reports the result to the user and the number of times the die has been rolled.
Some sample outputs are shown here.

```
Number rolled: 1
So far, you have: 1
Number rolled: 4
So far, you have: 1 4
Number rolled: 3
So far, you have: 1 3 4
Number rolled: 1
So far, you have: 1 3 4
Number rolled: 5
So far, you have: 1 3 4 5
Number rolled: 6
So far, you have: 1 3 4 5 6
Number rolled: 3
So far, you have: 1 3 4 5 6
Number rolled: 2
Congratulations, you have taken 8 rolls to get one of each
number between 1 and 6.
```

Part 2

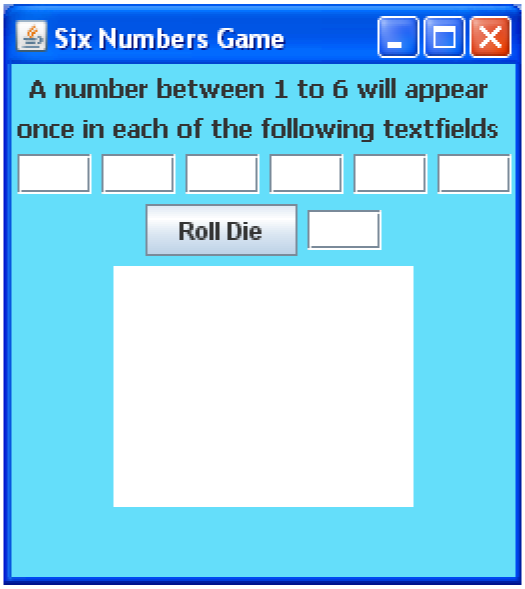
You need two other classes, class DisplaySixNumbersPanel and class SixNumbersPanel to create a GUI as shown below.

The class SixNumbersPanel has 6 textfields of 3 characters to display each of the six numbers (when the number is rolled the first time), a label for instructions, a button to roll the die and a textfield of 3 characters to display the current face value of the die and a textarea to display the number of rolls taken to obtain 6 unique numbers from 1 to 6.
This class also creates an object of class Player that checks and reports in the textfield, each time a unique number is rolled.

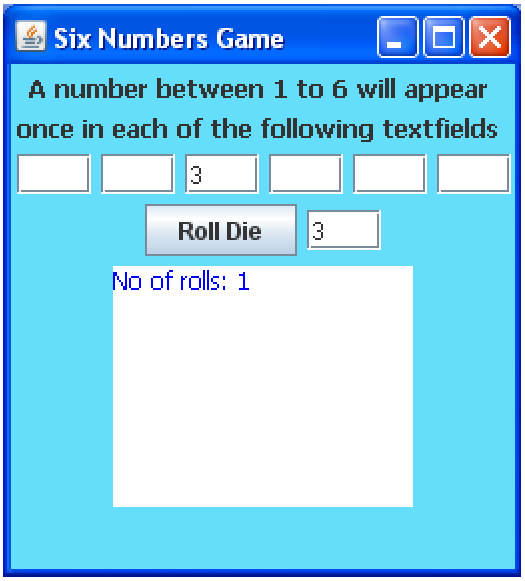
The code for class DisplaySixNumbersPanel is given.

```
import javax.swing.*;
public class DisplaySixNumbersPanel {
        public static void main(String[ ] args) {
                JFrame w1 = new JFrame("Six Numbers Game");

                SixNumbersPanel myPanel =new SixNumbersPane l( ) ;
                w1.add(myPanel);
                w1.setSize(260, 350);
                w1.setVisible(true);

                w1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        } //end main
} //end class
```

Some sample inputs and outputs for **Part 2** are shown following.
BEFORE   button click   AFTER first button click



A number between 1 to 6 will appear once in each of the following textfields



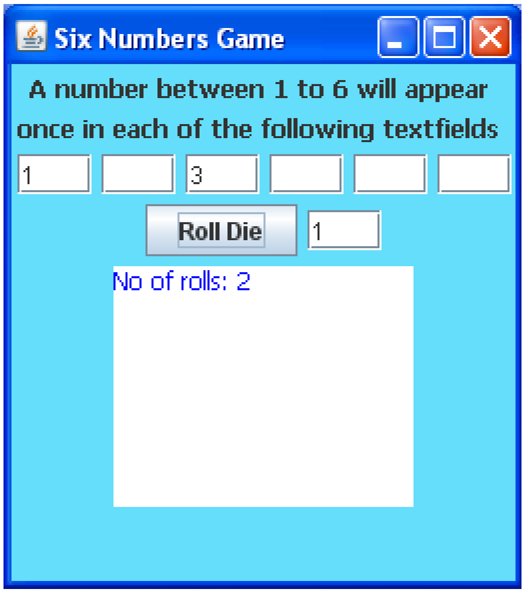
A number between 1 to 6 will appear once in each of the following textfields

3

3

No of rolls: 1

AFTER second button click



Six Numbers Game

A number between 1 to 6 will appear once in each of the following textfields

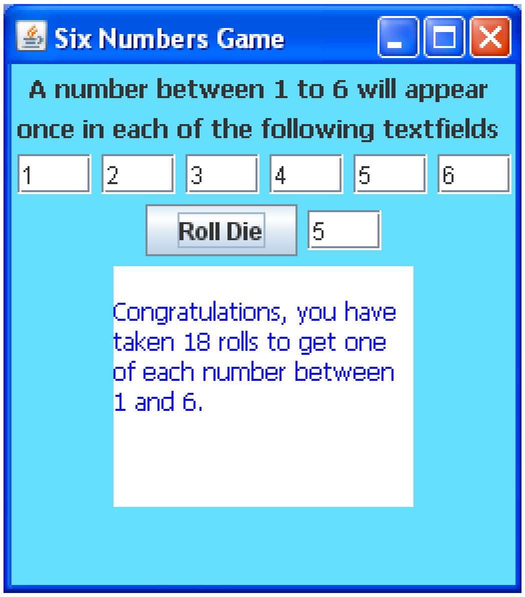
1 | | 3 | | |

1

No of rolls: 2

Continue rolling the die until all six numbers are obtained →

AFTER numerous button clicks and finally,



Six Numbers Game

A number between 1 to 6 will appear once in each of the following textfields

1 | 2 | 3 | 4 | 5 | 6

5

Congratulations, you have taken 18 rolls to get one of each number between 1 and 6.

<u>Part 3</u>

You are encouraged to enhance the basic product in Part 2 by allowing 2 players to play the *Unique Six* game. Each player alternately rolls a die and the game is over when one of the players obtains a set of six unique numbers from 1 to 6; report who the winner is and the total number of rolls taken to win the game. You may consider having a "New Game" button to play another game. If you make any modifications to class SixNumbersPanel, rename it as class SixNumbersPanel2.

Submission Details and Marking Scheme

It is compulsory that you demonstrate your work to your lab instructor on the due date and submit an <u>electronic version on VUCollaborate by or before the due date (Sunday 27 May 2018, 5 pm)</u>. Email submissions will <u>not</u> be accepted.
Assignment demonstration (20 minutes) is scheduled during the Swotvac week on 28-29 May 2018. During the demonstration, you will be asked questions on your work and modify/update code too. If you are unable to answer or modify/update code during the demo, you may get 0 for the assignment.
Appointment booking must be done via VU collaborate. The Appointment Booking link will be opened in Week 10.

If you are working in a group, you must complete a group formation form (only one submission per team) to your tutor by Wednesday May 2, 2018 1pm.

*** late submission will NOT be accepted***

Students to work in groups of two.
Groups MUST submit **<u>ONE</u>** copy of the assignment for both students. Each copy must be clearly labeled with both students' names and student id.

Academic Integrity

All work must be the group's own. Any unauthorised collaboration or copying will result in no credit for the assignment and possible overall failure in the subject.

What you have to hand in?

An electronic copy of

1. readme.doc file with information on how to run your program. Include any extra information about your design including UML class diagrams and program that you wish the marker to know.
2. Summary of tasks allocations – who did what.
3. A word document with evidence of trial runs of your program, i.e. screen printouts of the results where you have tested all the features of your code.
4. Code for all the classes that has been compiled and is ready to run. (jGRASP will be used to run and test your application)

   Each class should be fully documented commencing with a heading. In particular, the heading should include your **name, course, unit code, assignment details, date written** and a brief description of the class. At the start of each method, there should be a comment clearly describing what the method does. If you have a **partner, their name and student number** is should be clearly identified in the heading beside your name

NIT2112 – Assignment

Student ID	Student Name

Marking Guide

Description	Marks Received
Part 1 Application - 10 marks	
1. 3 classes as per the specification (OO design and coding) a. Player (4 marks) b. Die (2 marks) c. TestSixNumbers (4 marks)	
Part 2 GUI - 10 marks	
1. Main GUI -SixNumbersPanel (4 marks) 2. Text field being populated correctly – (2 marks) 3. Roll button working and reporting correctly (2 marks) 4. Clear display of results in text area (2 marks)	
Part 3 GUI application for 2 players – 10 marks	
1. Main GUI –(4 marks) 2. Fully working version allowing players to play in turn– (4 marks) 3. A new game button – (2 marks)	
Documentation and presentation – 10 marks	
1. UML diagrams for all classes – (4 marks) 2. Proper comments, Java programming conventions – (4 marks) 3. Full Testing and validation with print out of trial runs – (2 marks)	
Overall presentation – 10 marks	
1. Class demo – (5 marks) 2. Readme.doc file – (2 marks) 3. Originality and user friendliness – (3 marks)	
Total marks 50	
Out of 15	
Comments	