TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI

VUW VICTORIA UNIVERSITY OF WELLINGTON

Student ID: .....

EXAMINATIONS — 2014

TRIMESTER 2

SWEN222

Software Design

Time Allowed: THREE HOURS

Instructions: Closed Book.
There are 180 possible marks on the exam.

Answer all questions in the boxes provided.
Every box requires an answer.
If additional space is required you may use a separate answer booklet.

No calculators permitted.
Non-electronic Foreign language to English dictionaries are allowed.

No reference material is allowed.

Question	Topic	Marks
1.	Design patterns 1	30
2.	Functional design	30
3.	Design by Contract	30
4.	Software Design Qualities	30
5.	Design Patterns 2	30
6.	Refactoring	30
	Total	180

Student ID: .....

Question 1. Design Patterns 1

[30 marks]

(a) [8 marks] Provide a *class diagram* which describes the COMPOSITE pattern.

Consider the following description for describing *areas* and *regions* in a geographical application:

> "An *area* is a square section of land with dimensions measured in Kilometres (km^2). A *region* contains one or more areas or sub-regions. For example, a country can be considered as a region containing counties or states, which themselves are regions. An important concern is whether or not a given point (x, y) is contained within a region."

(b) [4 marks] Provide a *class diagram* for describing regions and areas.

(c) [10 marks] Provide a *Java implementation* for describing regions and areas.

(d) Consider the following additional requirements regarding regions. For each, briefly discuss whether or not this is true of your implementation.

(i) [4 marks] The structure of regions represents a tree.

(ii) [4 marks] A region cannot be contained in a region more than once.

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 2. Functional Design [30 marks]

Consider the following class for representing *binary data*.

```
1
2  public class BitSet {
3      private boolean[] data;
4
5      public BitSet(boolean[] data) {
6          this.data = data;
7      }
8
9      public boolean get(int index) {
10         return data[index];
11     }
12
13     public void set(int index, boolean bit) {
14         if(index >= data.length) {
15             // Make sure there is enough space
16             boolean[] nData = new boolean[data.length * 2];
17             System.arraycopy(data,0,nData,0,data.length);
18             data = nData;
19         }
20         data[index] = bit;
21     }
22 }
```

(a) [5 marks] An important aspect of the *functional programming paradigm* is that methods are *side-effect free*. Briefly, state what this means.

(b) For each of the following `BitSet` methods, briefly discuss whether or not it is side-effect free.

(i) [2 marks] `BitSet.get(int)`

(ii) [2 marks] `BitSet.set(int, boolean)`

(c) Another important aspect of the functional programming paradigm is *immutability*.

(i) [4 marks] Briefly, discuss whether or not the `BitSet` class is *immutable*.

(ii) [4 marks] Briefly, discuss the following statement:

"*Immutable classes can only have methods which are side-effect free.*"

(d) [8 marks] Rewrite the `BitSet` class to use a functional design.

(e) [5 marks] Using the `BitSet` class as an example, briefly discuss why programs using a functional design typically have fewer software bugs.

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 3. Design by Contract [30 marks]

Consider the following class which is used as part of an application for *matrix multiplication*.

```
1  public class Matrix {
2      private int[] items;
3      private int width;
4      private int height;
5
6      public Matrix(int width, int height) {
7          this.items = new int[width * height];
8          this.width = width;
9          this.height = height;
10     }
11     public int width() { return width; }
12     public int height() { return height; }
13
14     public int get(int x, int y) {
15         return items[x + (y * width)];
16     }
17     public void set(int x, int y, int item) {
18         items[x + (y * width)] = item;
19 }   }
```

(a) [2 marks] Briefly, state what a *pre-condition* is.

(b) [2 marks] Briefly, state what a *post-condition* is.

(c) [2 marks] Briefly, state what a *class invariant* is.

(d) For each of the following methods, given appropriate *pre-* and *post-conditions*.

(i) [4 marks] `Matrix.Matrix(int width,int height)`

(ii) [4 marks] `Matrix.get(int x,int y)`

(iii) [4 marks] `Matrix.set(int x, int y, int item)`

(e) [4 marks] Given an appropriate *class invariant* for the `Matrix` class.

(f) An important problem is to ensure the pre-condition of a method is respected by its callers and, similarly, that a method guarantees its post-condition holds. A simple solution is to use *runtime assertions*.

(i) [4 marks] Briefly, discuss how you would modify the `Matrix` class to use runtime assertions.

(ii) [4 marks] Unfortunately, runtime assertions cannot guarantee a program meets its specification. Briefly, discuss why not.

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 4. Software Design Qualities [30 marks]

(a) Coupling is an important concept relevant to software design.

(i) [4 marks] Define what is meant by the term *coupling* in software design.

(ii) [5 marks] Describe the *positive* implications of *strongly coupled* object oriented designs.

(iii) [5 marks] Describe the *negative* implications of *strongly coupled* object oriented designs.

(b) Inheritance is an important concept relevant to software design.

(i) [6 marks] Describe the benefits of *inheritance* in Java software designs.

(ii) [10 marks] Describe two limitations of *inheritance* in Java software designs and suggest alternative design approaches that address those limitations.

1)

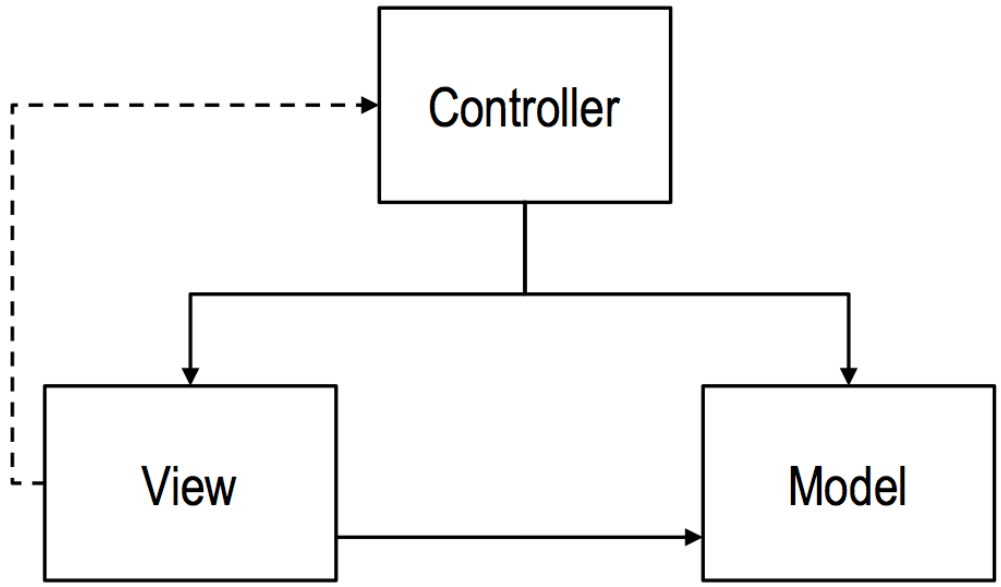
2)

Question 5. Design Patterns 2 [30 marks]

You have been asked to design a Java GUI implementation of the popular board game *Scrabble*. Key features include:

1. The game is played on a board with 15 squares on each side.
2. Players have hands of seven tiles, drawn from a pool of 100 tiles.
3. The first player places a word in the middle of the board, and subsequently every player in turn places letters on the board such that they create words linked to the existing words.
4. All words, including those created by intersecting tiles, must be legal words listed in the Scrabble dictionary.
5. Scores are calculated based on the letters used and the squares occupied by the letters.
6. Some of the squares modify the score calculated for letters or words placed on them, others do not.
7. When all of the tiles in the pool have been drawn, the game is over.
8. For this simple implementation, all players use the same computer, taking turns to use the mouse and keyboard as needed.

You are required to use the Model/View/Controller design pattern in your design:



(a) [6 marks] What features of this game would be implemented in the Model, View and Controller sub-systems?

Model:

View:

Controller:

(b) View Design

(i) [2 marks] Identify a Design Pattern that could be effectively used to implement key elements of the View's graphical user interface design.

(ii) [4 marks] Provide a UML class diagram summarising the key features of the design pattern identified in **(b)(i)**.

(iii) [6 marks] Describe how using that pattern helps create a more effective design for the View.

(c) Model Design

(i) [2 marks] Identify a Design Pattern that could be effectively used to implement key elements of the Model's design.

(ii) [4 marks] Provide a UML class diagram summarising the key features of the design pattern identified in **(c)(i)**.

(iii) [6 marks] Describe how using that pattern helps create a more effective design for the Model.

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 6. Refactoring [30 marks]

Consider the following classes defined as part of the design of a simple online adventure game:

```
1  public class GameBoard implements KeyListener {
2     public enum SquareType {EMPTY, WALL, CAVE, WATER};
3     public GameSquare squares[][];
4     private static int max_x = 50;
5
6     /** load the squares from our configuration **/
7     void setup (BufferedReader gameInfo) throws IOException {
8        String line = null;
9        int x = 0;
10       int y = 0;
11       while((line = gameInfo.readLine()) != null) {
12          squares[x][y] = new GameSquare();
13          squares[x][y].location
14             = new Point(x,y);
15          squares[x][y].occupied = false;
16          switch (line) {
17             case "Wall":
18                squares[x][y].sq = SquareType.WALL;
19                break;
20             case "Cave":
21                squares[x][y].sq = SquareType.CAVE;
22                break;
23             case "Water":
24                squares[x][y].sq = SquareType.WATER;
25                break;
26             default:
27                squares[x][y].sq = SquareType.EMPTY;
28          }
29          x++;
30          if (x > max_x){x = 0;y++;}
31       }
32    }
33
34    /** draw the board **/
35    void drawSquares() { ... }
36
37    /** handle user commands to move squares **/
38    public void keyTyped(KeyEvent e){ ... }
39    public void keyPressed(KeyEvent e){ ... }
40    public void keyReleased(KeyEvent e){ ... }
41 }
```

```
1  public class GameSquare {
2     public Boolean occupied;
3     public Point location;
4     public GameBoard.SquareType sq;
5     public Monster mIS1;
6     public Monster mIS2;
7     public Monster mIS3;
8     public Treasure tIS1;
9     public Treasure tIS2;
10    public Treasure tIS3;
11 }
12
13 public class Monster {
14    public String desc;
15    public Point pos;
16    public int hp;
17 }
18
19 public class Treasure {
20    public String desc;
21    public Point pos;
22    public int val;
23 }
```

(a) [18 marks] *Refactor* this design to improve it in *three significant and distinct ways*. For method bodies, you need only provide details needed to explain your refactorings. You can define new classes but only provide outlines of the methods for those classes.

(b) [12 marks] Identify the three significant changes you have made to this design, and for each explain how it has improved the design.

1)

2)

3)

****

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.