TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS – 2016

TRIMESTER 2

COMP 102
INTRODUCTION TO COMPUTER PROGRAM DESIGN

Time Allowed: TWO HOURS

CLOSED BOOK

Permitted materials: Silent non-programmable calculators or silent programmable calculators with their memories cleared are permitted in this examination.

Printed foreign language–English dictionaries are permitted.

No other material is permitted.

Instructions: Attempt ALL Questions.

The exam will be marked out of 120 marks.

Brief Java Documentation will be provided with the exam script

Answer in the appropriate boxes if possible — if you write your answer elsewhere, make it clear where your answer can be found.

There are spare pages for your working and your answers in this exam, but you may ask for additional paper if you need it.

Questions

	Marks
1. Understanding Java	[20]
2. Fields, Constructor and Methods	[25]
3. Files	[15]
4. Defining Interfaces	[12]
5. ArrayLists of Objects	[23]
6. Arrays	[25]

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 1. Understanding Java **(20 marks)**

(a) (6 marks) Calling a method and loops.

What will the following printNumbers method print out?

```
public void printNumbers(){
    int m = 2;
    this.printIt(m, m+10);
    UI.println(m);
}

public void printIt(int x, int y){
    while (x < y){
        x = x + 3;
        UI.println("x: " + x);
    }
    UI.println("finally "+ x);
}
```

Write your answers here

(b) (6 marks) Creating objects and calling methods on objects.

Consider the following Account class:

```
class Account {
    private int points;

    public Account(int x){
        this.points = x;
    }
    public void updatePoints(int y){
        this.points = this.points + y;
    }
    public int getPoints(){
        return this.points;
    }
    public boolean checkPoints(){
        if ((this.points >3) && (this.points <9)){
            return true;
        }
        else {
            return false;
        }
    }
}
```

What will the following useObjects method print out?

```
public void useObjects(){
    Account s1 = new Account(5);
    Account s2 = new Account(12);

    s1.updatePoints(2);
    UI. println (s1.getPoints ());

    UI. println ("----------");

    if (s2.checkPoints())
        UI. println (s2.getPoints ());
    else
        UI. println ("not interesting");

    UI. println ("----------");

    int x = s1.getPoints ();
    s2.updatePoints(x);
    UI. println (s2.getPoints ());
}
```

```
Write your answers here


--------------------


--------------------
```

(c) (8 marks) Using 2D Arrays.

What will the following array2DTest method print out?

```
public void array2DTest(){
    String [][] names = new String[][]{{"sam", "linda", "bob", "tom"},
                                       {"lisa", "luke", "tim", "dan"},
                                       {"james", "john", "zoe", "zac"}};

    UI. println (names[0][1]);
    UI. println (names[2][2]);

    UI. println ("--------------------");

    for ( int i = 0; i < names.length; i++){
        for( int j = 0; j < names[0].length; j++){
            if (names[i][j ]. length()>3){
                UI. println (names[i][j ]);
            }
        }
    }
}
```

```
Write your answers here




-------------------------
```

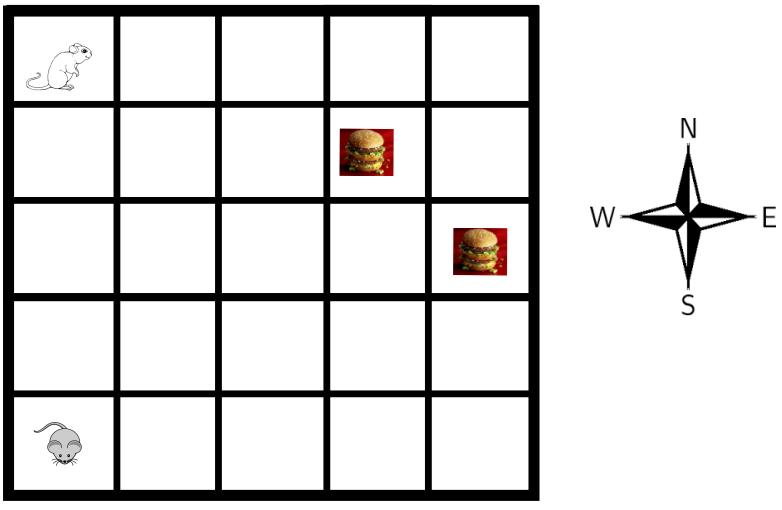
Question 2. Fields, Constructor and Methods **(25 marks)**

This question is about a small game with two mice looking for food on a 5*x*5 board. The board is shown below, with two cells (row 2, column 4) and (row 3, column 5) containing food.

The program has two classes. The first class MouseGame is given and your task is to implement the second class Mouse.

The first class MouseGame draws the board and the food, and uses the second class Mouse to create two mouse objects, and calls appropriate methods to direct them to look for food. One main method of this class is given below.

```
public void animate(){
  this.setUpBoard();                    //draws the board and food.
                                        //You do not need to define this method

  Mouse m1 = new Mouse(1,1, "east"); //create a mouse at row 1, column 1, facing east

  Mouse m2 = new Mouse(5,1, "south"); //create a mouse at row 5, column 1, facing south

  m1.move(3);                           //move three steps (three boxes) in the facing direction

  m1.turn("south");                     //turn to face south

  m1.move(1);

  m2.turn("north");
  m2.move(2);
  m2.turn("east");
  m2.move(4);
}
```



Complete the second class Mouse by defining data fields, the constructor and two methods turn and move, which are specified below.

Two extra methods draw and erase are also given below. They both use data fields, so make sure your data fields will work properly with these two methods.

Fields

```
// Store data
```

Constructor:

```
public Mouse(int row, int col, String dir)
/* Constructs a Mouse object and draws it on the board.
   The parameters are:
      the starting position: row number and column number,
      the direction the mouse is facing: one of "north", "south", "east" and "west".
*/
```

Methods:

```
public void turn(String newDir)
/* Changes the direction that the Mouse is facing.
   The parameter 'newDir' can be "north", "south", "east", or "west".
   This method should call other methods to erase the old mouse, and draw a new one.
*/

public void move( int step)
/* Makes the Mouse move in the facing direction for the specified number of steps.
   The parameter 'step' can be 1 to 4, and is the number of rows or columns the mouse moves.
   You should make sure that the mouse do not move out of the grid.
   This method should call other methods to erase the old mouse, and draw a new one.
*/


public void draw( ){
 /* Draws the mouse at the right position (row and col) on the board using the right image.
 This method is given below.
 */
 double x = 100 + (this.col-1) * this.size;
 double y = 100 + (this.row-1) * this.size;
 String name = "mouse-" + this.dir +".png";
 UI.drawImage(name, x+2, y+2, this.size-4, this.size-4);
}

 public void erase( ){
  /* Erases the mouse at the current position by erasing a square area.
  This method is given below.
  */
  double x = 100 + (this.col-1) * this.size;
  double y = 100 + (this.row-1) * this.size;
  UI.eraseRect(x+2, y+2, this.size-4, this.size-4);
}
```

```
public class Mouse{




  public Mouse(int row, int col, String dir){




  }
  public void turn(String newDir){




  }
  public void move(int step){
```

(More space is available. Please turn over the page)

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.

(To **continue**)

```
   }
}
```

Question 3. Files **(15 marks)**

This question is about a method that processes the index file of a cooking book. Each line of the file has a recipe name and some page numbers. The following is an example file that this method can process:

```
Apple Pie 8
Pizza 27, 30, 34-39
Hawaiian pizza 35, 36
Coconut Rice 120
```

Complete the searchIndex method on the facing page that prints the lines with the information of a particular recipe specified in the second parameter. The first parameter is the name of the file.

For the example file index.txt shown above, the method searchIndex("index.txt", "pizza") should print

Pizza 27, 30, 34-39
Hawaiian pizza 35, 36

The recipe match should not be case sensitive.

If a particular recipe is not found, it should print "not found".

Hint: the easiest way of doing this question is to read the file **line by line**.

```
    public void searchIndex(String filename, String recipeName){

        try{

    } catch(IOException e){UI.println("error: " + e);}
}
```

Question 4. Defining Interfaces **(12 marks)**

Suppose you are writing a program for a shop manager. The shop sells many different kinds of products (e.g. food, magazines, wine, etc.), which might have different features and may be sold in different ways (e.g. by kgs or by numbers). You are required to define a Product interface for different kinds of products, so all of them can be stored in an ArrayList of the Product type.

The following code is written for you. Product is an interface class.

```
public class Shop {
    ArrayList <Product> products = new ArrayList<Product>();

    public void printProducts(){
        for (Product p : products) {
            UI.println(p.getName());
            UI.println(p.getPrice());
        }
    }

    public void putOnSale(String name, double discount){
        for (Product p : products) {
            if (p.getName().equals(name)){
                p.setPrice(p.getPrice() * (1-discount));
                UI.println(p.toString());
            }
        }
    }
```

The printProducts method goes through all products and prints the name and price for each product. The putOnSale method changes the price of a particular product and prints the details of the product.

Define the Product interface, so that the code above would compile correctly.

```
public interface Product{




}
```

Question 5. ArrayLists of Objects **(23 marks)**

This question concerns a program for managing a collection of videos. The program has two classes: a VideoSite class and a Video class.

The Video class below defines Video objects, which store the video description, the length (minutes) of the video and the number of views.

```
class Video {
    private String description;
    private double minutes = 0;
    private int views = 0;

    public Video(String d, double m) {
        this.description = d;
        this.minutes = m;
    }

    public String getDes() {
        return this.description;
    }
    public double getMinutes() {
        return this.minutes;
    }
    public int getViews(){
        return this.views;
    }
    public void updateViews(int x) {
        this.views = this.views + x;
    }
    public String toString(){
        return (this.description +"(" + this.minutes + ")"+ " : " + this.views);
    }
}
```

(Question 5 continued on next page)

(Question 5 continued)

The VideoSite class defines a field to store the list of Video objects:

```
private ArrayList<Video> videos = new ArrayList<Video>();
```

You may assume that videos will never contain any null values.

The VideoSite class also defines several methods.

(a) (8 marks) Complete the following addNewVideo method that creates a new Video object and adds it to the end of the ArrayList. The parameters specify the description of the new video and the length in minutes of the video.

```
public void addNewVideo(String d, double m){




}
```

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 5 continued)

(b) (15 marks) Complete the following searchAndPrintPopular method that searches for videos whose description contain the search string specified in the parameter, and selects the most popular one to print out. The most popular video has the most views.

If there are not any videos that match the search string, it should print a "not found" message.

```
public void searchAndPrintPopular(String seachString){




}
```

Question 6. Arrays **(25 marks)**

This question is about a special integer array with only two types of values: zero or one, and the following is an example

```
private int[ ] nums = new int[ ]{1,1,0,0,0,1,1,1,1,1,1,0,0,1,0,0,0,0,1,1,1,0};
```

The example array looks like this:

1	1	0	0	0	1	1	1	1	1	1	0	0	1	0	0	0	0	1	1	1	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

(a) (8 marks) Suppose the following method is using the example array shown above. What will it print out?

```
public void testArray(){
   UI.println(this.nums[10]);
   UI.println(this.nums[1+2]);

   UI.println("-----------------");
   int count = 0;
   for(int i =0; i<this.nums.length; i++){
        if (this.nums[i]==1){
            UI.print(i + "  ");
            count++;
        }
   }
   UI.println();
   UI.println("-----------------");
   UI.println("count is " + count);
}
```

```
Write your answers here



--------------------



--------------------
```

(b) (17 marks) Complete the following findLongestSequenceOfOnes method that finds the longest sequence of 1s in the array. For example, using the example array shown on the facing page, this method should print

```
The length of the longest sequence of ones is : 6
start at index: 5
end at index: 10
```

The example array has many sequences of ones. This method should find the longest sequence and print its length and location (the starting index and the ending index). If there are multiple sequences with the same maximum length, print information for the first one.

```
public void findLongestSequenceOfOnes(){

}
```

* * * * * * * * * * * * * * *