

**EXAMINATIONS – 2014**
TRIMESTER 2**COMP 102**
INTRODUCTION TO
COMPUTER PROGRAM
DESIGN**Time Allowed:** THREE HOURS ***** **WITH SOLUTIONS** *******Instructions:** Closed Book
Attempt ALL Questions.

The exam will be marked out of 180 marks.

Silent non-programmable calculators or silent programmable calculators with their memories cleared are permitted.

Printed foreign language to English dictionaries are permitted.

Java Documentation will be provided with the exam script

No other material is permitted.

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. Call a method, if, while	[23]
2. Arrays and 2D Arrays	[35]
3. Files	[24]
4. Event driven input	[18]
5. Defining two classes	[25]
6. ArrayLists of Objects	[55]

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. Call a method, if, while

[23 marks]

(a) [6 marks] What will the following method1 print out?

```
public void method1(){
    int x = 10;
    this.method2(5);
    this.method2(x);
    UI.println ("x = " + x);
}
public void method2(int x){
    x = x+2;
    UI.println (x);
}
```

```
7
12
x= 10
```

(b) [8 marks] Complete the following `printNums` method to print all the odd numbers from 1 to 999, one number on each line, e.g.

1
3
5
7
...

```
public void printNums(){  
  
    int x = 1;  
    while (x <= 999){  
        UI.println (x + " ");  
        x = x + 2;  
    }  
  
}
```

(c) [9 marks] Complete the following `printMessage` method to print a message for the number given in the parameter. The number should be in the range `[0,100]`, and the method should print *Out of range* if it is not. The method should print *Good* for numbers in `[0, 70]`, *Better* for numbers in `[71,90]` and *Best* for numbers in `[91,100]`.

For example, `printMessage(75)` should print *Better*.

Note: `[a,b]` means any value in the `a` to `b` range, including `a` and `b`.

```
public void printMessage(int x){
    if (x>100||x<0)
        Ul.println ("Out of range");
    else if ( x <= 70 )
        Ul.print ("Good ");

    else if ( x <=90 )
        Ul.print ("Better ");

    else
        Ul.print ("Best ");
}
```

Question 2. Arrays and 2D Arrays

[35 marks]

(a) [8 marks] Consider the following `printArray` method which is passed an array of Strings.

```
public void printArray( String[ ] names){  
  
    names[1]=names[3]+ " to " + names[2];  
    UI.println (names[1]);  
    UI.println (names[3]);  
  
    for ( int i=0; i<names.length; i++){  
        if (names[i] != null){  
            UI.println (names[i]);  
        }  
    }  
}
```

What will the `PrintArray` method print out if passed the following array?

"Rome"	"Athens"	"Paris"	"London"	"Dublin"
0	1	2	3	4

```
London to Paris  
London  
Rome  
London to Paris  
Paris  
London  
Dublin
```

(Question 2 continued)

(b) [10 marks] Complete the following `printNumsArray` method which is passed an array of numbers. This method should print to the UI text pane all the numbers in order, one number per line, and also print the sum of the numbers at the end.

```
public void printNumsArray(int [ ] nums){  
    int total = 0;  
    for (int i=0; i<nums.length; i++){  
        UI.println (nums[i]);  
        total = total +nums[i];  
    }  
    UI.println ("the sum is " + total);  
}
```

(Question 2 continued)

(c) [7 marks] What will the following print2DArray method print out?

```
public void print2DArray(){
    int[ ][ ] nums = new int[ ][ ]{{3, 5, 7, 8},
                                     {6, 4, 11, 9},
                                     {0, 1, 2, 12}};

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

    for( int i = 0; i < nums.length; i++){
        UI.println (nums[i ][0]);
    }
}
```

5
2
3
6
0

(Question 2 continued)

(d) [10 marks] Complete the following `print2DArray` method which is passed a 2D array of numbers. This method should print to the UI text pane all the numbers in the 2D array as a table, one row per line.

```
public void print2DArray(int[ ][ ] nums){
    for( int r = 0; r < nums.length; r++){
        for( int c = 0; c < nums[0].length; c++){
            UI.print (nums[r][c ]);
        }
        UI.println ();
    }
}
```

Question 3. Files

[24 marks]

Suppose a weather forecast web site has forecast files containing data about the weather in the next week. Each line of a forecast file contains the day, weather description, the highest temperature and the lowest temperature in the day.

For example, the file "data.txt" contains the text:

```
Sunday rain 11 8
Monday cloudy 14 7
Tuesday clear 18 9
Wednesday clear 15 12
Thursday rain 19 10
Friday shower 17 11
Saturday clear 16 13
```

(a) [9 marks] Given the file above, what will the following method print out?

```
public void printData(){
    try{
        Scanner sc = new Scanner(new File("data.txt"));

        UI.println (sc.next ());
        String s = sc.nextLine ();
        if (sc.hasNext())
            UI.println (sc.next ());
        if (sc.hasNextInt())
            UI.println (sc.nextInt ());
        else
            s = s + sc.next ();
        UI.println (sc.nextInt ());
        s = s + sc.next ();
        UI.println (s);

        sc.close ();
    } catch(IOException e){UI.println("file reading failed"+e);}
}
```

s:

```
Sunday
Monday
14
rain 11 8Cloudy7
```

(Question 3 continued)

(b) [15 marks] Complete the following `printReport` method, whose parameter is the name of a forecast file. The `printReport` method should read the file and print out the clear days and the day with the maximum daily highest temperature.

For example, given the file on the previous page, `printReport("data.txt")` should print out

```
clear: Tuesday
clear: Wednesday
clear: Saturday
Hottest: Thursday
```

```
public void printReport(String filename){
    try{
        Scanner sc = new Scanner(new File(filename));
        int MaxHi = 0;
        String MaxDay = "";
        while (sc.hasNext()){
            String day = sc.next();
            String weather = sc.next();
            int hi = sc.nextInt();
            int lo = sc.nextInt();
            if (weather.equals("clear")){
                UI.println ("clear: " + day);
            }
            if ( hi > MaxHi)
                MaxHi = hi;
                MaxDay = day;
        }
        UI.println ("Hottest: " + MaxDay);

        sc.close ();
    } catch(IOException e){UI.println("file reading failed"+e);}
}
```

Question 4. Event Driven Input

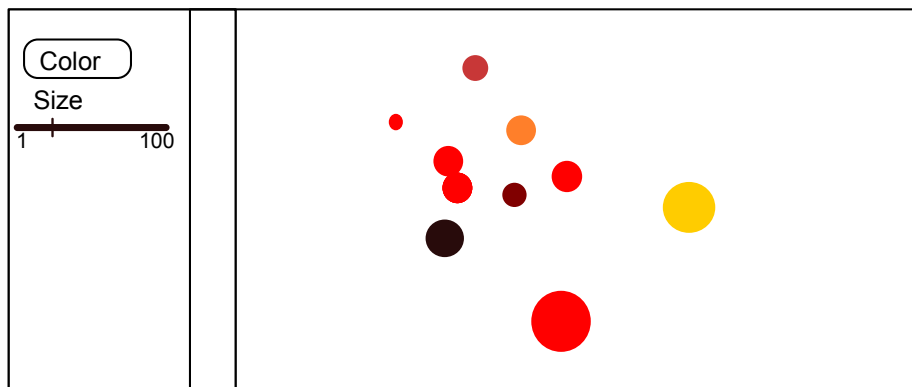
[18 marks]

Complete the Dots program on the facing page so that it allows the user to draw dots with different colors and in different sizes. The program should have a button and a slider, and also listen to the mouse.

When the color button is clicked, the color is set to a random color. The slider is for setting the size, which is the diameter of the future dots. The default size is 20.

When the user releases the mouse in the graphics pane, the program should draw a dot centered at the mouse position using the current color and size.

The diagram below shows what the program might draw. The dots should be in different colors.



(Question 4 continued)

```

                                UISliderListener, UIButtonListener, UIMouseListener
public class Dots implements {
    // Fields

    private double size =20;

    public Dots(){
        UI.setMouseListener(this);
        UI.addButton("Color", this);
        UI.addSlider("Size", 1, 100, 20, this);

    }

    public void buttonPerformed(String button){
        Color c = new Color((float)Math.random(), (float)Math.random(),(float)Math.random());
        UI.setColor(c);

    }

    public void sliderPerformed(String name, double v){
        this.size = v;

    }

    public void mousePerformed(String action, double x, double y) {
        if (action.equals("released")){
            UI.fillOval (x-this.size/2, y-this.size/2, this.size, this.size);
        }

    }

    public static void main(String[] arguments){
        new Dots();
    }
}

```

Question 5. Defining two Classes

[25 marks]

This question asks you to complete two classes `Person` and `CourseStaff` to store the contact details of people who are involved in teaching a course.

(a) [10 marks] Complete the `Person` class.

A `Person` object should contain two fields:

- name
- email

`Person` should have a constructor that takes a name and the email address and stores them in the two fields.

`Person` should have two methods:

- `setEmail`, which is passed the new email address and it can change the field to contain this value.
- `toString`, which returns a string containing the values of the two fields.

```
class Person{
    // fields
    private String name;
    private String email;

    // constructor
    public Person(String n, String e){
        this.name = n;
        this.email = e;
    }
    //methods
    public void setEmail(String e){
        this.email = e;
    }

    public String toString(){
        return this.name + " " + this.email;
    }
}
```

(Question 5 continued)

(b) [15 marks] Complete the `CourseStaff` class.

A `CourseStaff` object should contain three fields:

- `course`: the name of the course
- `lecturer`: a `Person`
- `tutors`: a list of `Persons`.

The constructor has been written for you.

`CourseStaff` should have three methods:

- `setLecturer`, which is passed a name and an email address. It should create a `Person` object and stores it in the field.
- `addTutor`, which is passed a name and an email address. It should create a `Person` object and add it to the list of tutors.
- `toString`, which returns a string with the course name and the names and emails of the lecturer and all the tutors.

```

public class CourseStaff{
    private String course;

    private Person lecturer;
    private ArrayList<Person> tutors = new ArrayList<Person>();

    public CourseStaff(String courseName){
        this.course = courseName;
    }

    public void setLecturer(String name, String email){

        this.lecturer = new Person(name, email);
    }

    public void addTutor(String name, String email){

        this.tutors.add(new Person(name, email));
    }

    public String toString(){

        String s = this.course + " " + this.lecturer.toString();
        for(Person p: this.tutors){
            s = s + " " + p.toString();
        }
        return s;
    }
}

```

Question 6. ArrayLists of Objects

[55 marks]

This question concerns a program for managing a database of houses listed by a real estate agent. Two of the classes in the program are a `House` class and a `HousesForSale` class.

The `House` class below defines `House` objects, which store information about individual houses.

```
public class House{
    private String suburb;
    private int size;
    private double price;

    public House(String sub, int sz, double p){
        this.suburb =sub;
        this.size = sz;
        this.price = p;
    }
    public String getSuburb(){
        return this.suburb;
    }
    public int getSize(){
        return this.size;
    }
    public double getPrice(){
        return this.price;
    }
    public String toString(){
        return this.suburb+ " " + this.size + " $" + this.price;
    }
}
```

(Question 6 continued)

(a) [7 marks] What will the following testHouse method print out?

```
public static void testHouse(){
    House h1 = new House("Karori",3, 400000);
    House h2 = new House("Kelburn", 4, 1000000);
    Ul.println (h1.getPrice ());
    Ul.println (h2.getSuburb());
    if (h1.getSize()>2){
        Ul.println (h1.toString ());
    }
}
```

```
400000.0
Kelburn
Karori 3 $400000.0
```

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 continued)

The HousesForSale class defines a field to store the database of House objects, and several methods, including `testArrayList`, `addHouse`, `searchHouse` and `sortInPriceOrder`.

```
private ArrayList<House> listings = new ArrayList<House>();
```

Assume that listings will never contain any null values.

(b) [8 marks] What will the following `testArrayList` method print out?

```
public void testArrayList () {
    House d = new House("Karori", 2, 250000);
    this.listings .add(d);
    this.listings .add(new House( "Aro Valley", 4, 500000));
    this.listings .add(0, new House( "Northland", 2, 300000));
    Ul. println (this.listings .get(0).getPrice ());
    Ul. println (this.listings .get(1).toString ());
    for (House h : this.listings ) {
        Ul. println (h.getSuburb());
    }
}
```

```
300000.0
Karori 2 $250000.0
Northland
Karori
Aro Valley
```

(c) [10 marks] Complete the following `addHouse` method that makes a new House object using the information given in the parameters, and adds it to the database. This method should add the house at the front of the list, and then display the first three houses on the list.

```
public void addHouse(String suburb, int size, double price){
    this.listings .add(0, new House(suburb, size, price));
    for (int i=0; i<3; i++){
        Ul. println (this.listings .get(i).toString ());
    }
}
```

(Question 6 continued on next 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.

(Question 6 continued)

(d) [15 marks] Complete the following `findHouses` method that asks the user for a suburb, a size (number of bedrooms), a minimum price and a maximum price, and then searches the database for all houses that match the suburb and the size and are in the price range. This method should print the details of all houses that meet the conditions, and print a not found message if no suitable houses are found.

```
public void findHouses(){
    int count=0;
    String s = UI.askString("Suburb: ");
    int b = UI.askInt("size: ");
    double p1 = UI.askDouble("Min Price: ");
    double p2 = UI.askDouble("max Price: ");
    for(House h: this.listings ) {
        if (s.equals(h.getSuburb())&&
            b==h.getSize()&&
            h.getPrice()>=p1&&
            h.getPrice()<=p2){
            UI.println (h.toString ());
            count++;
        }
    }
    if (count==0)
        UI.println ("not found");
}
```

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 continued)

(e) [15 marks] Complete the following `sortInPriceOrder` method that puts the houses on the list in increasing price order. Two houses with the same price can be ordered in either way. For example, suppose the houses are originally in this order

```
"Aro Valley",    3, $300000
"Brooklyn",     2, $500000
"Churton Park", 4, $300000
"Karori",       1, $150000
```

After the sorting, the houses should be in the following order

```
"Karori",       1, $150000
"Aro Valley",   3, $300000
"Churton Park", 4, $300000
"Brooklyn",     2, $500000
```

```
public void sortInPriceOrder(){

    ArrayList<House> newList = new ArrayList<House>();
    double minP = Double.MAX_VALUE;
    int minIndex = 0;

    int k = listings .size ();

    for( int i = 0; i < k; i++){

        for( int j = 0; j < listings .size (); j++){
            double p= listings .get(j).getPrice ();
            if (p < minP){
                minP = p;
                minIndex = j;
            }
        }
        newList.add( listings .get(minIndex));
        listings .remove(minIndex);
        minP=Double.MAX_VALUE;
    }
    this. listings = newList;
}

/* note: a better way is to swap the objects rather than remove,
so you do not destroy the original arraylist and
you do not need a new arraylist .
But this version is easier to understand
*/
}
```

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.
