

EXAMINATIONS – 2023

TRIMESTER 1

COMP 102/112  
INTRODUCTION TO  
COMPUTER PROGRAM  
DESIGN

**Time Allowed:** TWO HOURS

**CLOSED BOOK \*\*\*\*\* WITH SOLUTIONS \*\*\*\*\***

**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:**

- |                          |      |
|--------------------------|------|
| 1. Understanding Java    | [27] |
| 2. Design a class        | [15] |
| 3. Files                 | [23] |
| 4. ArrayLists of Objects | [40] |
| 5. 2D Arrays             | [15] |

**June 8, 2023**

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 1. Understanding Java****[27 marks]****(a) [4 marks] Calling methods.**

What will the following printStuff method print out?

```
public void printStuff (){
    Ul.println ("printing stuff");
    int x = 2;
    int y = 5;
    int z = y * 2;
    Ul.println ("z=" + z);
    z = this.myMethod(x, y);
    Ul.println ("x=" + x + " y="+ y + " z=" + z);
}

public int myMethod(int m, int n){
    m = m + n;
    m = m - 2;
    return m;
}
```

```
printing stuff
z=10
x=2 y=5 z=5
```

**(Question 1 continued)**

Consider the following Invoice class, specifying objects with two fields, a constructor and three methods.

---

```
class Invoice {
    private String company;
    private double value;

    public Invoice(String n, double p){
        this.company = n;
        this.value = p;
    }

    public void event1(){
        this.print();
        this.value = this.value - 10;
    }

    public void event2(double d){
        this.value = this.value + d;
        this.print();
    }

    public void print(){
        Ul.println(this.company + " : " + this.value);
    }
}
```

---

**(Question 1 continued)****(b) [6 marks] Calling methods on objects.**

Given the Invoice class on the facing page, what will the following useObjects method print out?

The boxes on the right are for your working if that helps.

```
public void useObjects() {
    Invoice i1 = new Invoice("KLM", 100);
    Invoice i2 = new Invoice("VUW", 30);

    i1.event1();
    i2.event2(10);
    Ul.println ("-----");
    i1.event2(50);
    i2.event1();
    Ul.println ("-----");
    i1.print();
    i2.print();
}
```

i1:

i2:

Write your answer here

```
KLM :100.0
VUW :40.0
```

```
-----
KLM :140.0
VUW :40.0
```

```
-----
KLM :140.0
VUW :30.0
```

(Question 1 continued on next page)

**(Question 1 continued)****(c) [8 marks] Arrays.**

What will the following arrayTest method print out?

```
public void arrayTest(){
    int [ ] nums = new int [ ]{2, 4, 5, 3, 1, 2, 8, 2};

    int n = 2;
    Ul. println (nums[2]);
    Ul. println (nums[n+3]);
    Ul. println (nums.length);
    Ul. println ("-----");

    for( int i = 0; i < 3; i++){
        nums[i] = nums[i] + nums[i+1];
        Ul. println (nums[i]);
    }
}
```

5

2

8

-----

6

9

8

**(Question 1 continued)****(d) [9 marks] ArrayLists.**

Complete the following average method that will return the average of the double values stored in values.

For example, if values stores [1.0, 5.0, 9.0] average returns 5.0.

```
public double average( ArrayList<Double> values) {
```

```
    double result = 0;
```

```
    for(double value : values) {  
        result += value;  
    }
```

```
    result = result / values.size ();
```

```
    return result ;
```

```
}
```

**Question 2. Design a class****[15 marks]**

Suppose you are writing a program to simulate a balloon game, where balloons are moved from the bottom to the top side of the screen. Each balloon is represented as an oval on the screen.

Complete the Balloon class on the facing page to specify Balloon objects.

- The fields should store the position, width, height and color of the balloon. Each balloon should have the default start position at (500, 600), where 500 is the centre of the x location of the balloon and 600 is the y position of the centre of the balloon.
- Declare the fields so they cannot be accessed directly outside the class.
- The constructor should create a Balloon object with the specified width, height, and color. It should call the draw method to show the balloon at the default start position.
- The draw method should draw a filled oval for the Balloon at the right location using the right color and size.
- The move(double pixels) method should move the balloon upwards for the specified number of pixels. It should not redraw it.
- The resetBalloon method should reset the balloon to the default position, if the balloon has reached the top of the screen (i.e. if any part of the balloon is above the screen).



```
public class Balloon {  
  
    private double x = 500;  
    private double y = 600;  
    private double width;  
    private double height;  
    private Color color;  
    public Balloon(double w, double h, Color c) {  
  
        this.width = w;  
        this.height = h;  
        this.color = c;  
        this.draw();  
    }  
    public void draw(){  
  
        UI.setColor( this.color );  
        UI.fillOval( this.x-this.width/2, this.y-this.height/2, this.width, this.height );  
  
    }  
    public void move(double s){  
  
        this.y = this.y - s;  
  
    }  
    public void resetBalloon(){  
  
        if (this.y < this.height/2) this.y = 600;  
  
    }  
}
```

**Question 3. Files****[23 marks]**

A file is shown below. Each line contains a name and some double numbers. The number of doubles might be different from line to line and there might be multiple lines for one name.

```
Kurt 2.5 5.3
Tui 3.1 3.5
Liz 4.3 1.0 0.4
Kurt 6.8
Tui 4.1
```

(a) **[10 marks]** What will the following `printFromFile` method print out if passed the file-name of the file given above and the name "Tui"?

```
public void printFromFile( String filename, String name){
    ArrayList<String> lines = readAllLines( filename);
    for( String str : lines) {
        Scanner line = new Scanner(str);
        if( line .hasNext()) {
            String person = line .next ();
            String numbers = line.nextLine ();
            if (!person.equals(name)) {
                UI. println (numbers);
            }
        }
    }
}

public ArrayList<String> readAllLines( String fname){
    ArrayList<String> ans = new ArrayList<String>();
    try {
        Scanner scan = new Scanner(Path.of(fname));
        while (scan.hasNext()){
            ans.add(scan.nextLine ());
        }
        scan. close ();
    }
    catch(IOException e){UI. println ("File reading failed");}
    return ans;
}
```

```
2.5 5.3
4.3 1.0 0.4
6.8
```

(b) [13 marks] Complete the following personMinimum method that will read the file, finds the minimum number for the person, specified in the parameter, name, and prints the minimum of the numbers.

If the same name is in the file multiple times, it refers to the same person.

If the name is not included in the file, show a "not found" message.

You must call the readAllLines method on the facing page to read all the data from the file, and then process the data line by line using an ArrayList.

The name of the data file is data.txt.

```
public void personMinimum(String name){

    ArrayList<String> lines = readAllLines("data.txt");
    double min = Double.MAX_VALUE;
    boolean infile = false;

    for (String line : lines) {
        Scanner sc = new Scanner(line);

        if(sc.next().equals(name)) {
            infile = true;
            while(sc.hasNextDouble()) {
                double num = sc.nextDouble();
                if(num<min) {
                    min = num;
                }
            }
        } else {
            sc.nextLine();
        }
    }

    if( infile )
        Ul.println(name + " minimum: " + min);
    else
        Ul.println(name + " not found");

}
```

**Question 4. ArrayLists of Objects****[40 marks]**

This question concerns a program for scheduling meetings.

The program has two classes: a Scheduler class and a Meeting class.

The Meeting class below defines Meeting objects, which store the date and hour of the meeting.

The date is a string, such as "Monday W1", referring to the Monday in the first week of the year. (Note the space between Monday and W1)

The hour is stored as an int and should have a value of 9, 10, 11, 12, 13, 14, 15, or 16 (meetings can only be scheduled during work hours).

---

```
class Meeting {
    private String date;
    private int hour;

    public Meeting(String d, int h) {
        this.date = d;
        this.hour = h;
    }

    public String getDate(){
        return this.date;
    }

    public int getHour(){
        return this.hour;
    }
}
```

---

(Question 4 continued on next page)

**(Question 4 continued)**

(a) [15 marks] Complete the `searchByWeek(...)` method that will search the schedule to print the information for all meetings that occur in the week specified by the parameter. For example, `searchByWeek("W15")` should print the date and hour for every meeting that occurs in Week 15, e.g.

W15 Monday @ 14

W15 Tuesday @ 11

W15 Friday @ 12

If no matching meetings are found, this method should display a "no meetings" message.

Valid weeks start at "W1" and end at "W52".

```

public class Scheduler {

    private ArrayList<Meeting> meetings;

    /**
     * Uses a scanner to populate meetings
     */
    public Scheduler(Scanner sc) {
        //Code omitted
    }

    public void searchByWeek(String week){

        boolean found = false;

        for(Meeting m : meetings) {
            Scanner sc = new Scanner(m.getDate());
            String day = sc.next();
            String meetingWeek = sc.next();

            if(meetingWeek.equals(week)) {
                found = true;
                UI.printf("%s %s @ %d\n", week, day, m.getHour());
            }
        }

        if(!found) {
            UI.println("no meetings");
        }

    }
}

```

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.

(b) [15 marks] Complete the `addMeeting(...)` method that will create a meeting with the date and hour specified by the parameters, and then add the meeting to the schedule **if and only if** it does not conflict with an existing meeting.

Two meetings conflict if they occur on the same date, at the same hour. You must ensure that only valid meetings are added (weeks and hours). You can assume that the day of the week is correct.

The method should return `true` if the meeting was successfully added, or `false` otherwise.

```
public boolean addMeeting(int week, String day, int hour ){

    boolean toAdd = true;

    String date = day + " W" + week;

    for(Meeting m : meetings) {
        if(m.getDate().equals(date) && m.getHour() == hour) {
            toAdd = false;
        }
    }

    if(toAdd && week > 0 && week < 53 && hour > 8 && hour < 17) {
        meetings.add(new Meeting(date, hour));
        return true;
    } else {
        return false;
    }

}
```

(c) [10 marks] Meetings are stored in files followed with the names of participants (each person has a unique name in the file), e.g.

W15 Monday @ 14 Karsten Marcus

W15 Tuesday @ 11 Marcus Ghassem Karsten Monique

W15 Friday @ 12 Monique Ghassem

Each line is one meeting, followed by the participants of each separate meeting.

Complete the findMeetings(...) method that will return an ArrayList of all the meetings where the provided person is a participant.

For example, if sc is a Scanner of the above file, findMeetings(sc, "Karsten") should return an ArrayList consisting of these two meetings:

W15 Monday @ 14

W15 Tuesday @ 11

```

public ArrayList<Meeting> findMeetings(Scanner sc, String name) {
    ArrayList<Meeting> result = new ArrayList<Meeting>();

    while(sc.hasNextLine()) {
        String line = sc.nextLine();
        if (line.contains(name)) {
            Scanner lineSc = new Scanner(line);
            String week = sc.next();
            String day = sc.next();
            sc.next();
            String date = day + " " + week;
            int hour = sc.nextInt();

            for(Meeting m : meetings) {
                if(m.getDate().equals(date) && m.getHour() == hour) {
                    result.add(m);
                }
            }
        }
    }

    return result;
}

```



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 5. 2D Arrays****[15 marks]**

The gameData field is declared to hold a 2D array as follows:

```
private int [ ][ ] gameData;
```

Each row stores the final score from a rugby game. The first value is the score by the home team, and the second value stores the score by the away team.

(a) **[5 marks]** If gameData is assigned the following data:

```
this.gameData = new int[ ][ ]{{37, 10},  
                                {22, 15},  
                                {19, 19},  
                                {6, 21},  
                                {14, 12}};
```

What will the following method print out?

```
public void printNums(){  
    UI.println (this.gameData[0][1]);  
  
    for (int j=0; j<5; j++){  
        UI.println (this.gameData[j][0]);  
    }  
  
}
```

Write your answers here

```
10  
37  
22  
19  
14
```

(Question 5 continued on next page)

**(Question 5 continued)**

(b) [10 marks] Complete the following `homewinStats` method that will calculate and print the percentage of games where the home team has won.

For example, if `gameData` contains the data in part (a), `homewinStats` should print out the following numbers:

Wins: 60.00%

If no games have been played, print "No games played".

You may assume that each row in the array has precisely two columns.

```

public void homewinStats(){

    int homewins = 0;
    for( int i = 0; i < gameData.length; i++) {
        if(gameData[i][0] > gameData[i][1]) homewins++;
    }

    if(gameData.length>0) {
        double percentage = 100*(((double)homewins)/gameData.length);
        Ul. printf ("Wins:%6.2f%%\n",percentage);
    }
    else
        Ul. println ("No games played");
//or
    double homewins = 0;
    for( int i = 0; i < gameData.length; i++) {
        if(gameData[i][0] > gameData[i][1]) homewins++;
    }

    if(gameData.length>0) {
        double percentage = 100*(homewins/gameData.length);
        Ul. printf ("Wins:%6.2f%%\n",percentage);
    }
    else
        Ul. println ("No games played");
}

```

\*\*\*\*\*