

EXAMINATIONS – 2018

TRIMESTER 1

COMP 102  
INTRODUCTION TO  
COMPUTER PROGRAM  
DESIGN

**Time Allowed:** TWO HOURS \*\*\*\*\* WITH SOLUTIONS \*\*\*\*\*

**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.

**June 29, 2018**  
**Questions:**

- Understanding Java [25]
- GUI programs [18]
- Files [22]
- 2D Arrays [20]
- ArrayLists of Objects [35]

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****[25 marks]****(a) [8 marks] Calling methods.**

What will the following printStuff method print out?

```
public void printStuff (){
    Ul.println ("printing stuff");
    int x = 10;
    int y = 20;

    int z = this.myMethod(x, y);

    Ul.println ( this.myMethod(y, x));

    Ul.println ("x=" + x + " y=" + y + " z=" + z);
}
public int myMethod(int m, int n){
    Ul.println ("m=" + m + " n=" + n);
    int t = m - n;
    return t;
}
```

```
printing stuff
m=10 n=20
m=20 n=10
10
x=10 y=20 z=-10
```

**(Question 1 continued)**

Consider the following Account class, specifying objects with two fields and three methods.

---

```
class Account {
    private String title ;
    private double balance;

    public Account(String n ){
        this.title = "AC:" + n;
    }

    public void check(){
        this.report ();
        this.balance = this.balance - 5;
    }
    public void invest (double amt){
        this.balance = (this.balance + amt) * 10;
        this.report ();
    }

    public void report (){
        Ul.println (this.title + " $" + this.balance);
    }
}
```

---

**(Question 1 continued)****(b) [9 marks] Calling methods on objects.**

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

```
public void useObjects(){
    Account ac1 = new Account("Jim");
    Account ac2 = new Account("Jan");

    ac2.report ();

    ac1.invest (100);
    ac2.invest (50);
    Ul.println ("-----");

    ac1.invest (80);
    ac2.check();
    Ul.println ("-----");

    ac1.check();
    ac2.report ();

}
```

ac1:

ac2:

Write your answers here

```
AC:Jan $0.0
AC:Jim $1000.0
AC:Jan $500.0
```

```
-----
AC:Jim $10800.0
AC:Jan $500.0
```

```
-----
AC:Jim $10800.0
AC:Jan $495.0
```

(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, 6, 3, 4, 1, 2, 9, 0};

    int n = 0;
    UI.println (nums[n] + 3);
    UI.println (nums[n+3]);
    UI.println (nums[nums.length-1]);

    UI.println ("-----");
    for (int i=1; i < nums.length-1; i++) {
        if (nums[i] > nums[i-1] && nums[i] > nums[i+1]) {
            UI.println (nums[i]);
        }
    }
}

```

5  
4  
0

-----  
6  
4  
9

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. GUI programs****[18 marks]**

Complete the following DrawCircleGUI program that allows the user to draw a filled circle on the graphics pane and then press a button to make it grow.

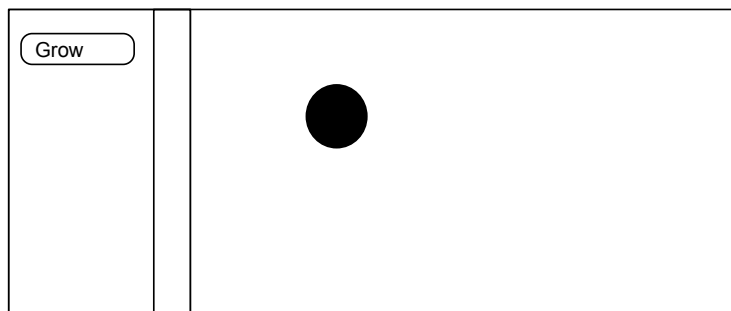
When the user clicks the mouse on the graphics pane, the program will clear the graphics pane and draw a filled circle of diameter 30 centered at the clicked place.

When the user presses the Grow button, the diameter of the circle will grow by 20 and the circle will be redrawn at the same place (the center does not change).

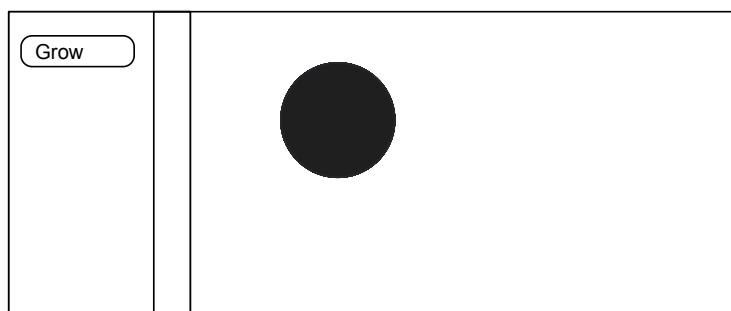
You need to

- declare the fields,
- complete the setupGUI method to add a button and a mouse listener.
- complete the doMouse method
- complete the grow method

The interface should look like this after the user clicks the mouse



The circle should grow after the user presses the button





```
public class DrawCircleGUI {
    private double centerX;
    private double centerY;
    private double size;

    public DrawCircleGUI(){
        this.setupGui();
    }
    public void setupGUI(){
        UI.addMouseListener( this :: doMouse);
        UI.addButton("Grow", this::doGrow);
    }
    public void doMouse(String action, double x, double y) {
        if (action.equals("clicked")){ // or "released"
            UI.clearGraphics ();
            this.size = 30;
            this.centerX = x;
            this.centerY = y;
            UI.fillOval (this.centerX-this.size/2, this.centerY-this.size/2,
                        this.size, this.size );
        }
    }

    public void grow() {
        this.size = this.size + 20;
        UI.clearGraphics ();
        UI.fillOval (this.centerX-this.size/2, this.centerY-this.size/2,
                    this.size, this.size );
    }
}
```

**Question 3. Files****[22 marks]**

This question concerns statistical data about the area of farmland used for different crops, taken from a database of Māori owned farms.

- The first line contains column titles: "Year" and four kinds of crops.
- Each following line contains a year, and four numbers specifying the area planted in each kind of crop.

The file is shown below:

Year	KiwiFruit	Avocados	WineGrapes	Onions
2007	311.44	39.8	75.9	212
2009	344.95	39	83	230
2011	349.38	49.9	207.5	290
2012	347.18	114.61	298.7	291
2014	314.27	78.9	408	317.6

(a) **[8 marks]** What will the following printFromFile method print out if passed the name of the file given above?

```
public void printFromFile( String filename){
    try{
        Scanner sc = new Scanner(new File(filename));
        UI.println (sc.next ());
        sc.nextLine ();
        String s = sc.next ();
        double d = sc.nextDouble();
        UI.println (d);
        UI.println (s);
        UI.println (sc.next ());
    }
    catch(IOException e){UI.println ("Fail: " + e);}
}
```

Write your answers here

```
Year
311.44
2007
39.8
```

(b) [14 marks] Suppose we have a field containing a 2D array, initialised as follows:

```
private double[ ][ ] cropData = new double[5][4];
```

Complete the following loadFarmDataToArray method that will read the file, and load the data on farmland areas into the array in the cropData field.

- Each row should have the area for a particular year
- Each column should have the area for a particular crop

After calling this method, cropData should contain the following array:

311.44	39.8	75.9	212
344.95	39	83	230
349.38	49.9	207.5	290
347.18	114.61	298.7	291
314.27	78.9	408	317.6

Assume that the file has 6 rows and 5 columns, as shown on the previous page.

```
public void loadFarmDataToArray(String filename){
    try{
        Scanner sc = new Scanner(new File(filename));

        sc.nextLine();
        for (int r=0; r<5; r++){ // or int r = 0; while(sc.hasNextLine()){
            sc.next();
            for (int c=0; c<4; c++){
                this.cropData[r][c] = sc.nextDouble();
            }
        } // or r++; }
        sc.close();

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

**Question 4. 2D Arrays****[20 marks]**

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

```
private double [ ][ ] cropData;
```

Please note that this question does not depend on Question 3.

(a) **[8 marks]** If cropData is assigned the following value:

```
this.cropData = new double[ ][ ]{{311.44, 39.8, 75.9, 212},
                                   {344.95, 39.0, 83.0, 230},
                                   {349.38, 49.9, 207.5, 290},
                                   {347.18,114.6, 298.7, 291},
                                   {314.27, 78.9, 408.0, 317}};
```

What will the following method print out?

```
public void printNums(){
    UI.println ( this .cropData [0][2]);

    for ( int j=0; j<4; j++){
        UI.println ( this .cropData[2][j ]);
    }
}
```

Write your answers here

```
75.9
349.38
49.9
207.5
290.0
```

(Question 4 continued on next page)

**(Question 4 continued)**

(b) [12 marks] Complete the following `columnAverages` method that will calculate and print the average of each column in the 2D array in the `cropData` field.

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

333.4 64.4 214.6 268.1

Your method should still work well when the number of rows and columns are not 5 and 4. You may assume that the array contains at least one row and at least two columns.

```
public void columnAverages(){  
  
    for (int c=0; c<this.cropData[0].length; c++){  
        double tot = 0;  
        for (int r=0; r<this.cropData.length; r++){  
            tot = tot + cropData[r][c];  
        }  
        UI.print (tot/this.cropData.length + " ");  
    }  
  
}
```

**Question 5. ArrayLists of Objects****[35 marks]**

This question concerns a program for managing phone books. The program has two classes: a PhoneBookManager class and a Contact class.

The Contact class below defines Contact objects, which store details about names and phone numbers.

---

```
public class Contact {  
    private String name;  
    private int num;  
  
    public Contact(String name, int num) {  
        this.name = name;  
        this.num = num;  
    }  
  
    public String getName() {  
        return this.name;  
    }  
  
    public int getNum() {  
        return this.num;  
    }  
}
```

---

(Question 5 continued on next page)

**(Question 5 continued)**

The PhoneBookManager class declares a field to store the list of Contact objects:

```
private ArrayList<Contact> phoneBook = new ArrayList<Contact>();
```

(a) [8 marks] What will the following method print out?

```
public void testArrayList (){
    Contact a = new Contact("lisa", 4763237);
    Contact b = new Contact("linda", 4638328);
    this.phoneBook.clear();
    this.phoneBook.add(new Contact("Sam", 3473800));
    this.phoneBook.add(a);
    this.phoneBook.add(new Contact("Tom", 7942984));
    Ul.println (this.phoneBook.get(0).getName());
    Ul.println (this.phoneBook.contains(b));
    Ul.println (this.phoneBook.indexOf(a));
    Ul.println ("-----");
    Ul.println (this.phoneBook.size());
    Ul.println (this.phoneBook.get(2).getNum());
    this.phoneBook.set(2,a);
    Ul.println ("-----");
    for(Contact f: this.phoneBook){
        Ul.println (f.getName());
    }
}
```

Write your answers here

```
Sam
false
1
```

```
-----
3
7942984
```

```
-----
Sam
lisa
lisa
```

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) [12 marks] Complete the following `searchByName` method that will search the phone-Book to print the information for all people whose names contain the String specified in the parameter. For example, `searchByName("li")` should print name and number for everyone whose name contains "li", e.g.

linda : 6438328

annalie : 7643237

If no matching names are found, this method should display a "not found" message.

```
public void searchByName(String pattern){

    boolean exist = false;
    for (Contact p : this.phoneBook){
        if (p.getName().contains(pattern)){
            Ul.println (p.getName()+ ": " + p.getNum());
            exist = true;
        }
    }
    if (! exist){
        Ul.println ("not found");
    }

}
```

(c) [15 marks] Complete the following cleanUp method that will clean up the phoneBook by removing duplicates and conflicting Contacts.

- When a duplicate Contact (same name and same number as a previous Contact) is found, print a “duplicate removed” message, and remove it.
- When a conflicting Contact (same name but different number as a previous Contact) is found, print a message with the name and both numbers, and remove the conflicting Contact.

**Hint:** You may create a new ArrayList to store the revised list, and then assign it back to phoneBook.

```
public void cleanUp(){
    ArrayList<Contact> newBook = new ArrayList<Contact>();
    for (Contact f: phoneBook){
        String name = f.getName();
        int num = f.getNum();
        boolean keep = true;
        for (Contact p: newBook) {
            if ((p.getName().equals(name)) && (p.getNum()==num)) {
                keep = false;
                Ul.println ("duplicate removed");
            }
            else if (p.getName().equals(name)){
                keep = false;
                Ul.println (name + ":" + num + " " + p.getNum());
            }
        }
        if (keep) {
            newBook.add(f);
        }
    }
    phoneBook = newBook;
}
```

\*\*\*\*\*