



EXAMINATIONS — 2012

END-OF-YEAR

**COMP 102
INTRODUCTION TO
COMPUTER PROGRAM
DESIGN**

Time Allowed: 3 Hours

Instructions: Attempt ALL Questions.

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

The exam will be marked out of 180 marks.

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

Non-electronic foreign language dictionaries are permitted.

Java Documentation will be provided with the exam script.

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	[75]
2. Objects, Arrays of Objects	[25]
3. Interface Classes	[20]
4. 2D Arrays	[30]
5. Files, Array of Numbers	[30]

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

[75 marks]

(a) [8 marks] What will the following fragment of Java code print out?

```
int x = 0;
int y = 0;
while (x < 10){
    UI.println (x + " : " + y);
    y = y + 1;
    x = x + 2;
}
UI.println ("done");
```

(b) [8 marks] The conditionalTest method below has two parameters and prints out a message.

```
public void conditionalTest(int first , int second){
    String ans = " ";

    if ( first > 2) {
        ans = ans + "start";
    }

    if ( first > 3 && second < 10) {
        ans = ans + " first";
    }
    else if ( first < 2 || second > 12) {
        ans = ans + " second";
    }
    else {
        ans = ans + " all";
    }
    UI.println (ans + " done");
}
```

What would the following calls to conditionalTest print out?

conditionalTest(2, 8); \Rightarrow

conditionalTest(1, 12); \Rightarrow

conditionalTest(5, 5); \Rightarrow

(c) [10 marks] Complete the following smallNumber method which returns the minimum of the two numbers specified by the two parameters (doubles). For example, smallNumber(4.5, 9.1) should return 4.5.

```
public      smallNumber
```

```
}
```

(d) [12 marks] Consider the following printNames method.

```
public void printNames(String [ ] names){
    Ul.println (names[3]);
    Ul.println (names.length);
    Ul.println (names[0].length());
    Ul.println (names[1] + names[2+1]);

    for( int i=0; i<names.length-1; i++){
        names[i] = names[i+1];
    }
}
```

Suppose that the variable names is defined as follows:

```
String [ ] names = new String[ ]{"Peter","James","Ann","Jill","Paul","Neil","John"};
```

names:	Peter	James	Ann	Jill	Paul	Neil	John
	0	1	2	3	4	5	6

What will be printed in the text pane if printNames(names) is called?

Show the contents of names after the method call

```
printNames(names);
```

names:							
	0	1	2	3	4	5	6

(Question 1 continued on next page)

(Question 1 continued)

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 continued on next page)

(Question 1 continued)

(e) [12 marks] Suppose the file `data.txt` contains the text:

chocolate	3.40	tim tam	17
plain	1.90	ginger nuts	20
chocolate	2.80	mallowpuffs	10

What will the following `printBiscuits` method print out?

```
public void printBiscuits (){
    try{
        Scanner scan = new Scanner (new File("data.txt"));
        UI.println (scan.next ());
        UI.println (scan.nextDouble());
        UI.println (scan.nextLine ());

        while ( scan.hasNext() ){
            if (scan.hasNextDouble()){
                scan.nextDouble();
            }
            else {
                UI.println (scan.next ());
            }
        }
        scan.close ();
    }
    catch (IOException e){UI.println("File reading failed");}
}
```

(Question 1 continued on next page)

(Question 1 continued)

(f) [13 marks] Complete the `House` class on the facing page which stores information about houses for sale.

A `House` object should contain three fields:

- `address`, which contains the address of the house.
- `size`, which contains the number of bedrooms (*e.g.* 3)
- `price`, which contains the price.

`House` should have a constructor that takes one `String` parameter, one integer parameter and one double parameter and stores them in the fields.

`House` should have three methods:

- `getPrice`, which returns the price of the house.
- `setPrice`, which updates the `price` field with a specified new price.
- `toString`, which returns a `String` containing the values of all the fields, *e.g.*
`"21 karori, 3 brms, $650000.0"`

(Question 1 continued on next page)

(Question 1 continued)

```
public class House{
```

```
    // fields
```

```
    // constructor
```

```
    // methods
```

```
}
```

(Question 1 continued on next page)

(Question 1 continued)

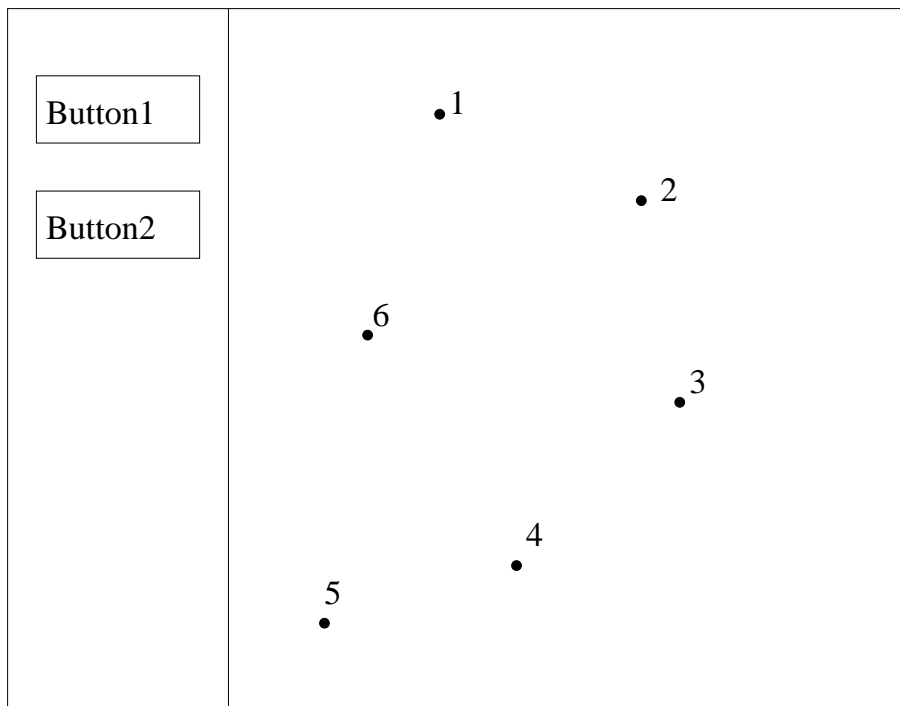
(g) [12 marks]

Consider the `PatternDrawer` class on the facing page, which constructs a simple GUI. It has two fields that store a position (`lastX`, `lastY`) and a field to store the current shape to draw.

Sketch below what the program would draw on the canvas if the user took the following actions in sequence:

1. press mouse at point 1
2. release mouse at point 2
3. press the "Button1" button
4. press mouse at point 3
5. release mouse at point 4
6. press the "Button2" button
7. press mouse at point 5
8. release mouse at point 6

Draw your answers in the figure below.



(Question 1 continued on next page)

(Question 1 continued)

```

public class PatternDrawer implements UIButtonListener, UIMouseListener{

    private double lastX = -1;
    private double lastY = -1;
    private String shape = " ";

    public PatternDrawer(){
        UI.setMouseListener(this);
        UI.addButton("Button1", this);
        UI.addButton("Button2", this);
        UI.addButton("Clear", this);
    }

    public void buttonPerformed(String button){
        if (button.equals("Button1"))
            this.shape = "shape1";
        if (button.equals("Button2"))
            this.shape = "shape2";
        if (button.equals("Clear"))
            UI.clearGraphics();
    }

    public void mousePerformed(String action, double x, double y) {
        if (action.equals("pressed")){
            this.lastX = x;
            this.lastY = y;
        }
        else if (action.equals("released")){
            if (this.shape.equals("shape1")){
                UI.drawOval(x-10, y-10, 20, 20);
            }
            else{
                UI.drawLine(this.lastX, this.lastY, x, y);
            }
        }
    }
}

```

Question 2. Objects, Arrays of Objects

[25 marks]

This question concerns a `FlatFinder` program to let users examine a list of available flats. The `FlatFinder` class (on the facing page) stores the information about the flats in a field containing an array of `Flat` objects. It also has a `count` field that contains the number of `Flat` objects, and the `Flat` objects are stored in cells 0 through `count-1` of the array.

The `Flat` class, for representing information about individual flats, is shown below. You should use the appropriate `Flat` methods in your answers.

```
public class Flat{
    // fields
    private String address;
    private String landlord;
    private int bedrooms; // number of bedrooms
    private double rent; // weekly rent
    // constructor
    public Flat(String addr, String landl, int beds, double rnt){
        this.address = addr;
        this.landlord = landl;
        this.bedrooms = beds;
        this.rent = rnt;
    }
    //methods
    public void printDetails (){
        UI.printf ("%d beds, $%.0f /week, %s (%s)\n",
            this.bedrooms, this.rent, this.address, this.landlord );
    }
    public String getAddress() {
        return this.address;
    }
    public boolean hasLandlord(String name){
        return (this.landlord.equals(name));
    }
    public int getBeds() {
        return this.bedrooms;
    }
    public double getRent() {
        return this.rent;
    }
}
```

(Question 2 continued on next page)

(Question 2 continued)

The following are some of the fields of the FlatFinder class:

```
public class FlatFinder{  
  
    private Flat [ ] flats = new Flat[1000];  
    private int count = 0;
```

(a) [10 marks] What will the following testFlats method print out?

```
public void testFlats(){  
    Flat f = new Flat("1/12 Victoria", "Alex", 4, 400);  
    Flat g = new Flat("90 Glenmore", "Bob", 8, 600);  
  
    Ul.println (f.getRent());  
    g.printDetails ();  
  
    count= 0;  
    flats [count] = f;  
    count++;  
    flats [count]= g;  
    count++;  
  
    flats [0]. printDetails ();  
    Ul.println ( flats [1]. getBeds());  
}
```

(Question 2 continued)

(Question 2 continued on next page)

(Question 2 continued)

(b) [15 marks] Complete the following `costOfBiggest` method to return the rent of the biggest flat with the most number of bedrooms. If there is a tie for biggest, it should return the rent of the earliest one in the list. This method should also print the details of the biggest flat.

```
public double costOfBiggest(){
```

```
}
```

Question 3. Interface Classes

[20 marks]

This question addresses the design of a program to help a manager record all the products in a shop. There are a number of different kinds of products: fruits, vegetables, food, toys, etc. The different kinds of products behave differently, so there is a separate class for each. However, every product will have the following in common:

- fields to store values such as the name of the product, the barcode, the price and whether it is on sale.
- a `getInfo` method that returns a short String containing a description of the product, including the name, price and whether it is on sale.
- a `getPrice` method that returns the price of the product.
- a `putOnSale` method with a parameter to specify the discount, e.g. 30 for 30% off. This method should set a true value to the `onSale` field and update the price accordingly.

The program must have a single array containing all the products. To do this, we need an interface class called `Product` that specifies a type that includes the different kinds of products.

(a) [10 marks] Complete the following definition of the `Product` interface class.

```
public interface Product {
```

```
}
```

(Question 3 continued on next page)

(Question 3 continued)

(b) [10 marks] Modify/Extend the following Food class so that a Food is a type of Product, ensuring that it correctly implements the Product interface class.

```
public class Food implements Product{
    private String name;
    private String barcode;
    private double price;
    private boolean onSale = false;
    private String expiryDate;
    private String pictureName;

    public Food(String Nm, String code, double val, String d, String fileName){
        this.name = Nm;
        this.barcode = code;
        this.price = val;
        this.expiryDate = d;
        this.pictureName = fileName;
    }

    public void display(double x, double y){
        UI.drawImage(this.pictureName, x, y);
    }

    // Write your answers here

}
```

Question 4. 2D Arrays

[30 marks]

The Noughts and Crosses is a game consisting of a 3×3 grid of squares. The class stores the current state of the game in a field containing a 2D array of char values ('O' for noughts, 'X' for crosses, '\0' for empty):

```
public class NoughtsCrosses{  
    private char [ ][ ] board;  
    private sqSize = 50;  
    :
```

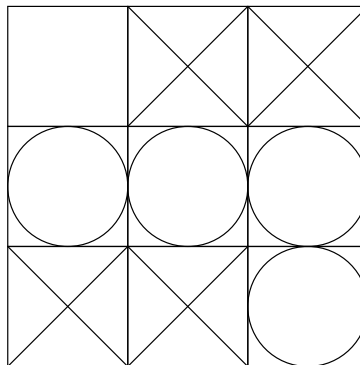
You are to write two methods for the NoughtsCrosses class.

(a) [15 marks] Complete the following drawBoard method which should draw the board in the Graphics pane.

For example, if the 2D array contains

```
{ { '\0', 'X', 'X' },  
  { 'O', 'O', 'O' },  
  { 'X', 'X', 'O' } }
```

it should draw the following diagram.



You should draw nine squares for the board. The top left square should be at position (100,100), and the width of the squares should be 50 units. Each square may contain a circle (for 'O' value), or two lines (for 'X' value) or nothing for a default value of '\0'(not a char).

```
private void drawBoard(){  
    UI.clearGraphics();
```

```
}
```

(Question 4 continued on next page)

(Question 4 continued)

(b) [15 marks] Complete the following `checkRow` method whose parameter is the index of a row on the board, and should return `true` if all the cells in the given row are filled with the same char (either 'O' or 'X'), and `false` otherwise.

For example, `checkRows(0)` on the game shown on the previous page should return `false` and `checkRows(1)` should return `true`.

```
public boolean checkRow(int row){
```

```
}
```

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. Files, Array of Numbers

[30 marks]

This question stores and processes a collection of integer numbers in an array. The array is declared and created as follows:

```
private int[ ] data = new int [200];
private int count = 0;
```

(a) [12 marks] Suppose the numbers.txt file contains about 100 integer numbers, and some text which may appear anywhere in the file. Complete the following loadNumbers method to read all the numbers into the data array and ignore all the text.

The file numbers.txt may look like this:

```
10 3 text may appear
anywhere 1 20 in 5
2 4 in the 10 file
```

```
public void loadNumbers() {
    try{
        Scanner scan = new Scanner(new File("numbers.txt"));

        scan.close();
    }
    catch(IOException e){UI.println("File reading failed");}
}
```

(Question 5 continued on next page)

(Question 5 continued)

(b) [18 marks] Write a method to merge two sorted/ordered list of numbers into one sorted list of numbers.

The result list should be saved in the array data field **data** (with the help of the integer field **count**). You may assume that the **data** array has enough extra space for all the numbers.

For example, given a sorted list of 5 numbers:

list1:	3	8	15	20	60	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9
count1:	5									

and an array of 4 sorted numbers:

list2:	2	4	8	12	0	0	0	0
	0	1	2	3	4	5	6	7
count2:	4							

`merge(list1, 5, list2, 4)` should produce the following array:

data:	2	3	4	8	8	12	15	20	60	0
	0	1	2	3	4	5	6	7	8	9
count:	9									

Note that the numbers in `list1` and `list2` are ordered and the numbers in the `data` array must be ordered too. Duplicate numbers are allowed.


```
public void merge(int [ ] list1 , int count1, int [ ] list2 , int count2){
```

```
}
```

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.