



**EXAMINATIONS — 2009**

**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.

Non-programmable calculators without a full alphabetic key pad 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.

**Questions**

	<b>Marks</b>
1. Understanding Java	[65]
2. Files	[22]
3. Arrays of Objects	[34]
4. 2D Arrays	[31]
5. Designing with Interface Classes	[28]

**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**

[65 marks]

(a) [6 marks] What will the following fragment of Java print out?

```

int x = 10;
double y = 2.5;
int temp = x;
x = (int) y;
y = temp;
x = x + x;
String z = y + " : " + x;
System.out.println(" x=" + x);
System.out.println(" y=" + y);
System.out.println(" z=" + z);

```

(b) [6 marks] Consider the following condition method (note the “ifs” and “elses” carefully) :

```

public int condition(int x, int y) {
    int ans = 0;
    if (x > 20){
        ans = ans + 2;
    }
    if (x > y){
        ans = ans + x;
    }
    else if (x < y) {
        ans = ans + y;
    }
    else {
        ans = ans + 4;
    }
    return ans;
}

```

What would the following calls to condition return?

```

condition(25, 10) ==>
condition(3, 6) ==>
condition(7, 7) ==>

```

(c) [6 marks] What will the following fragment of Java print out?

```
for (int j = 0; j < 4; j++){  
    System.out.println(j + " " + j*2);  
}  
System.out.println("Done");
```

(d) [6 marks] What will the following fragment of Java print out if the user enters the three words hello, you, and quit in response to the prompts. (Note the println and print carefully.)

```
Scanner sc = new Scanner(System.in);  
String word = "Start";  
while ( ! word.equals("quit") ){  
    System.out.println(word);  
    System.out.print("Enter a word: ");  
    word = sc.next();  
}  
System.out.println("Done");
```

(Question 1 continued on next page)

**(Question 1 continued)**

(e) [9 marks] Suppose the variable `numbers` is declared and initialised as follows:

```
int [ ] numbers = new int [ ] {2, 7, 9, 3, 5, 8};
```

numbers:	2	7	9	3	5	8
	0	1	2	3	4	5

What will the following code fragment print out? (Note the `println` and `print` carefully.)

```
System.out.println(numbers[3]);
System.out.println(numbers[0] + numbers[4]);
System.out.println(numbers[numbers[0]]);

for( int i = numbers.length-1; i >= 0; i--){
    System.out.print(numbers[i]+ " ");
}
```

(f) [6 marks] Suppose that the variable `numbers` is declared and initialised as before:

```
int [ ] numbers = new int [ ] {2, 7, 9, 3, 5, 8};
```

Show the contents of `numbers` after the following `change` method is called on `numbers`:  
`change(numbers)`.

```
public void change(int [ ] nms){
    for( int i = 0; i < nms.length-1; i++){
        nms[i] = nms[i+1];
    }
}
```

numbers:						
	0	1	2	3	4	5

(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)**

**(g)** [9 marks] Complete the following `scaleUp` method. `scaleUp` has one parameter – an array of `doubles` – and should scale up all the values in the array by 30%, e.g. changing 10.0 to 13.0. You may assume that the length of the array is greater than 0.

```
public void scaleUp (double[] data) {
```

```
}
```

**(h)** [9 marks] Complete the following definition of a `book` class. `book` objects should have two fields called `title` and `year` to specify the book title and the year published.

The class should have a constructor that takes two arguments and sets the fields to its arguments. The class should have one method called `toString` with no parameters that returns a `String` describing the object (which should contain the values of both fields).

```
public class Book {
```

```
}
```

(Question 1 continued on next page)

**(Question 1 continued)**

The following `MovingShape` class defines `MovingShape` objects, which have three fields, one constructor and two methods.

```
public class MovingShape{
    private DrawingCanvas canvas;
    private int radius = 10;
    private int x, y;

    public MovingShape(DrawingCanvas c, int xx, int yy){
        this.x = xx;
        this.y = yy;
        this.canvas = c;
        this.draw();
    }

    public void draw(){
        this.canvas.fillOval (this.x, this.y, this.radius*2, this.radius*2);
    }

    public void move(int dx, int dy){
        this.canvas.clearRect(this.x, this.y, this.radius*2+2, this.radius*2+2);
        this.x = this.x + dx;
        this.y = this.y + dy;
        this.draw();
    }
}
```



The following `ScreenSaver` class is the main class that will create `MovingShape` objects and call their methods to move the objects around.

(i) [8 marks] Complete the `firstMove` method in the `ScreenSaver` class. It should create two `MovingShape` objects at any two different places you choose, and then move the first one right by 20 pixels and move the second one down by 50 pixels.

```
public class ScreenSaver {

    private JFrame frame = new JFrame("Screen Saver");
    private DrawingCanvas canvas = new DrawingCanvas();

    public ScreenSaver(){
        this.frame.setSize(600,400);
        this.frame.getContentPane().add(this.canvas, BorderLayout.CENTER);
        this.frame.setVisible(true);
        this.firstMove();
    }

    public void firstMove(){

    }

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

## Question 2. Files

[22 marks]

The following file marking.txt is a simplified marking sheet for assignment 4.

```
Marking Sheet for COMP102 2009   Assignment 4
LoginName: xgao
Name: Xiaoying Gao
ID: 300140000
Marked by: Alex
```

```
-----
StockAnalyser Mark: 75
=====
```

```
-----
ScreenSaver Mark: 55
=====
```

(a) [10 marks] In the box below show what the following print method will print out, if called with the statement

```
print("marking.txt");
```

```
public void print( String fname) {
    try{
        Scanner fileScan = new Scanner(new File(fname));
        while (fileScan.hasNext()){
            if (fileScan.hasNextInt()){
                int num = fileScan.nextInt ();
                System.out.println(num);
            }
            else {
                String junk = fileScan.next ();
            }
        }
        fileScan.close ();
    }
    catch(IOException e){System.out.println("File reading failed: "+e);}
}
```

(Question 2 continued on next page)

**(Question 2 continued)**

**(b)** [12 marks] Complete the following `extractNameMarks` method so that it reads data from a file in the format described above, and prints out the name and the marks. On the example file above, it should print

```
Xiaoying Gao
75
55
```

```
public void extractNameMarks(String fname){
    try{
        Scanner fileScan = new Scanner(new File(fname));

        fileScan.close ();
    }
    catch(IOException e){System.out.println("File reading failed: "+e);}
}
```

### Question 3. Arrays of Objects

[34 marks]

This question concerns a program that lets a shop manager keep track of the products. The program uses an array of **Product** objects to store all the products the shop is selling. The shop manager can add a new product, delete a product and print lists of all the products etc.

The **Product** class (shown below) represents the information about individual products: the description, the price, and whether it is on sale.

```
public class Product{
    private String description;
    private double price;
    private boolean sale;

    public Product(String s, double d, boolean b){
        this.description = s;
        this.price = d;
        this.sale = b;
    }

    public String getDescription(){
        return this.description;
    }
    public double getPrice(){
        return this.price;
    }
    public void setPrice(double d){
        this.price = d;
    }
    public boolean onSale(){
        return this.sale;
    }
    public void printLabel() {
        System.out.print(this.description + " $" + this.price);
        if (this.sale)
            System.out.println("Special!");
        else
            System.out.println ();
    }
}
```

The **Shop** class uses the **Product** class, and contains an array to store the information about a collection of **Products**. The fields of the **Shop** class are declared and initialised as follows:

```
private final int maxItems = 200;
private Product[] items = new Product[maxItems];
private int count = 0;
```

(a) [10 marks] The following `testPrint` method in the `Shop` class shows how to assign values to the array and how to access the elements in the array. What will it print out to `System.out`?

```
public void testPrint () {  
    this.count =0;  
    items[this.count] = new Product("tomato", 3.4, true);  
    this.count++;  
    items[this.count] = new Product("potato", 2.5, true);  
    this.count++;  
    System.out.println (items [0].getDescription ());  
    items [1].printLabel ();  
    items [0].setPrice (2.0);  
    System.out.println (items [0].getPrice ());  
}
```

(b) [12 marks] Complete the following `addProduct` method in the `Shop` class which should allow a user to add a new product specified by the parameters to the collection. If the array is full, the method should print `The list is full` and not add a product. Please note that there is a field `count` to specify the number of products in the array.

```
public void addProduct (String s, double d, boolean b) {
```

```
}
```

(Question 3 continued on next page)

(Question 3 continued)

**(Question 3 continued)**

(c) [12 marks] Complete the following `salePriceUpdate` method in the `Shop` class so that it allows the user to update the price for all products that are on sale using a discount specified by the parameter. For example, if the discount is 0.3, then all sale products are for sale at 30% off the original price. This method should update the prices and print labels for all sale products.

```
public void salePriceUpdate(double discount) {
```

```
}
```

## Question 4. 2D Arrays

[31 marks]

The program shown below stores a matrix in a 2D array. Field declarations, the constructor and one method are given:

```
public class Matrix{

    private final int maxRows = 4;
    private final int maxCols = 4;
    private int[ ][ ] data = new int[maxRows][maxCols];

    public Matrix(int[ ][ ] d) {
        this.data = d;
    }

    public void printSomeData( ) {
        System.out.println(data [0][2]);
        System.out.println(data [2][0]);

        for ( int i = 0; i < maxRows; i++) {
            System.out.print(i + " : ");
            for ( int j = 0; j < maxCols; j++) {
                System.out.print(this.data[i][j] + " ");
            }
            System.out.println ();
        }
    }

    :
    :
}
```

The following is a test method in this class:

```
public static void main(String[] arguments){
    int[ ][ ] d = new int[ ][ ]
        {{10, 2, 3, 14},
         { 6, 8, 22, 1},
         { 7, 5, 4, 20},
         { 3, 1, 8, 9}};

    Matrix m = new Matrix(d);
    m.printSomeData();
}
```

It initialises the 2D array with the following matrix and calls the printSomeData method.

$$\begin{bmatrix} 10 & 2 & 3 & 14 \\ 6 & 8 & 22 & 1 \\ 7 & 5 & 4 & 20 \\ 3 & 1 & 8 & 9 \end{bmatrix}$$



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

(b) [12 marks] Complete the following `simplify` method so that it changes every number in the matrix to either one or zero based on the given threshold. If the number is bigger than the given threshold, then it is changed to one and otherwise to zero. For example `simplify(5)` should change the example matrix shown on the facing page to:

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

```
public void simplify ( int threshold){
```

```
}
```

(Question 4 continued on next page)

(Question 4 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 4 continued on next page)

**(Question 4 continued)**

(c) [11 marks] Complete the following `symmetricMatrix` method that will return true if the matrix saved in the 2D array is symmetric, and returns false otherwise.

Here is a definition of Symmetric Matrix given in the Wikipedia:

In linear algebra, a symmetric matrix is a square matrix,  $A$ , that is equal to its transpose  $A = A^T$ . The entries of a symmetric matrix are symmetric with respect to the main diagonal (top left to bottom right). So if the entries are written as  $A = (a_{ij})$ , then  $a_{ij} = a_{ji}$  for all indices  $i$  and  $j$ . For example, the following 3x3 matrix is symmetric:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & -5 \\ 3 & -5 & 6 \end{bmatrix}$$

```
public boolean symmetricMatrix(){
```

```
}
```

## Question 5. Designing with Interface Classes

[28 marks]

Suppose you are implementing a `Researcher` program for a staff member to keep a list of publications. You have already implemented the following two classes for representing information about the two kinds of publications:

```
public class JournalArticle implements Publication{

    private String paperTitle;
    private String journalName;
    private int vol;
    private int year;

    public JournalArticle(String title , String name, int vol, int year){
        this.paperTitle = title ;
        this.journalName = name;
        this.vol = vol;
        this.year = year;
    }

    public String getTitle (){ return this.paperTitle; }
    public int getYear(){ return this.year; }
    public void print(){
        System.out.printf("%s, %s, (%d), %d \n",
                           this.paperTitle, this.journalName, this.vol, this.year);
    }
}

public class ConferencePaper implements Publication{

    private String paperTitle;
    private String location;
    private int year;

    public ConferencePaper(String title, String loc, int y){
        this.paperTitle = title ;
        this.location = loc;
        this.year = y;
    }
    public String getTitle (){ return this.paperTitle; }
    public int getYear(){ return this.year; }
    public void print(){
        System.out.printf("%s, %s, %d. \n",
                           this.paperTitle, this.location, this.year);
    }
}
```

(a) [8 marks] Suppose you wanted to use an interface class to enable the `Researcher` program to store a collection of `JournalArticle` and `ConferencePaper` objects. Define an appropriate interface class `Publication`, specifying all the method headers.

```
public
```

```
}
```

(b) [8 marks] The `Researcher` class stores the collection of `Publications` in an array. The relevant fields in the `Researcher` class are:

```
private final int maxPublications = 20;
private Publication [] list = new Publication[maxPublications];
```

Suppose someone has written the following test method for the `Researcher` program. What will this method print out?

```
public void test1 () {
    list [0] = new ConferencePaper("AutoWrapper", "WI Hong Kong", 2003);
    list [1] = new JournalArticle("AraQ", "Artificial Intelligence", 3, 2008);
    list [2] = new JournalArticle("AutoWrapper", "WI IAT Journal", 4, 2007);
    for (int i = 0; i < list.length; i++) {
        if (list [i] != null) {
            list [i].print ();
        }
    }
}
```

(Question 5 continued on next page)

**(Question 5 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 5 continued)**

(c) [12 marks] Suppose some of the publications in your collection have the same title. You want to keep only one copy and delete other ones with the same title. Complete the `deleteSimilar` method of the `Researcher` class to do this. Please note that you are allowed null objects anywhere in your array.

```
public void deleteSimilar( ) {
```

```
}
```

\*\*\*\*\*