



## EXAMINATIONS — 2003

MID-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.

Non-programmable calculators without full alphabet keys are permitted.

Non-electronic foreign language dictionaries are permitted.

### Questions

	<b>Marks</b>
1. Understanding Java programs	[40]
2. Writing programs with “while” and “if”	[27]
3. Arrays	[38]
4. Recursion	[15]
5. Containers and genericity	[30]
6. Inheritance, events and Swing	[30]

**Question 1. Understanding Java programs**

[40 marks]

(a) [4 marks] What will be printed when the following method is called?

```
public void q1a() {  
    int x = 5;  
    int y = 10;  
    x = y;  
    System.out.println(x);  
    System.out.println(y);  
}
```

(b) [5 marks] What will be printed when the following method is called?

```
public void q1b() {  
    int x, y, z;  
    x = 4;  
    y = 2;  
    z = 10;  
    System.out.println("Sum1 is " + (x + y + z));  
    System.out.println("Sum2 is " + x + " + " + y + " + " + z);  
    System.out.println(y / x);  
    System.out.println(Math.sqrt(x) + Math.pow(x, y));  
    System.out.println(Math.max(Math.min(x, y), z));  
}
```

(c) [4 marks] Consider the following method definition:

```
public void q1c(int x) {
    if (x >= 0) {
        if (x <= 5)
            System.out.println("yellow");
        else
            System.out.println("blue");
    }
    else
        System.out.println("red");
}
```

(i) [2 marks] What will be printed when this method is called with **6** as its argument?

(ii) [2 marks] What will be printed when this method is called with **-2** as its argument?

(d) [6 marks] What will be printed when the following method is called?

```
public void q1d() {
    int min = 1;
    int max = 5;
    for (int i = min; i < max; i++) {
        System.out.println(i + " " + i * i);
    }
}
```

(e) [6 marks] What will be printed when the following method is called?

```
public void q1e() {
    String s = "message";
    int i = 0;
    while (i < s.length() - 1) {
        System.out.print(s.charAt(i));
        System.out.print(s.charAt(i + 1));
        System.out.println();
        i = i + 1;
    }
}
```

(f) [6 marks] What will be printed when the following method is called?

```
public void q1f() {
    String s = "This JAVA book costs $120";
    for (int i = 0; i < s.length(); i++) {
        if (s.charAt(i) >= 'a' && s.charAt(i) <= 'z')
            System.out.print("l");
        else if (s.charAt(i) >= 'A' && s.charAt(i) <= 'Z')
            System.out.print("u");
        else
            System.out.print(s.charAt(i));
    }
}
```

(g) [9 marks] Consider the following program:

```
public class NameProgram {
    public static void main(String[] args) {
        String name = "Robert";

        TestName nn = new TestName("John");
        nn.print1();
        nn.print2(name);
        nn.print3("Sharon", "CO 240");
        nn.print3("CO 240", "Sharon");

        TestName n = new TestName();
        n.print1();
        n.setName("peter");
        n.print1();

        System.out.println(name);
    }
}
class TestName {
    private String name = "Bob";

    public TestName(String n) {
        name = n;
    }
    public TestName() {
    }
    public void print1() {
        System.out.println(name);
    }
    public void print2(String name) {
        System.out.println(name);
    }
    public void print3(String myName, String myRoom) {
        System.out.println(myName + " " + myRoom);
    }
    public void setName(String x) {
        name = x;
    }
}
```

What will this program print out?

**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. Writing programs with “while” and “if”**

[27 marks]

**(a)** [13 marks] Programming with “while”

Complete the following program, so that it will read a line of text and print each word on a separate line. A “word” is any sequence of characters not containing a space. You may assume that words are only separated by a single space. The text should be read using `JOptionPane.showInputDialog`. You may assume that the input is non-empty, that is, the input contains at least one word. The output should be displayed using `System.out`.

For example, if the program reads “This is a line of text”, the program should print

```
This
is
a
line
of
text
```

```
import javax.swing.*;

public class WordSplitter {

    public static void main(String[] args) {
```

**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)** [14 marks] Programming with ‘if’

Write a program that prints the price for a travel ticket. There are three kind of tickets: oneway, return and 10 trip, and two kinds of buyers: adult and student.

The prices are as follows:

	oneway	return	10 trip
adult	\$2	\$4	\$16
student	\$1	\$2	\$8

Your program should prompt for a ticket type and a buyer type, and then print out a string consisting of the input and the price, for example, ‘adult oneway \$2’.

The input should be read using two calls on `JOptionPane.showInputDialog`. You may assume that the input is always correct. The output should be displayed using `System.out.println`.

```
import javax.swing.*;

public class TicketPrice {

    public static void main(String[] args) {
```

**Question 3. Arrays**

[38 marks]

**(a) [20 marks] 2D Array of Integers**

We have a class consisting of a data field `box` and four methods: `array1()`, `array2()`, `array3()`, and `array4()`. The data field is declared and created as a two-dimensional array of integers.

```
private int[][] box=
    { { 4, 6, 2, 1 },
      { 7, 8, 3, 9 },
      { 0, 4, 3, 1 },
      { 8, 5, 2, 9 } };
```

**(i) [3 marks]** What will the following method print out?

```
public void array1() {
    System.out.println(box[0][2]);
    System.out.println(box[2][0]);
    System.out.println(box[2][2]);
}
```

**(ii) [6 marks]** After executing the following method, what will the content of the file `out.txt` be?

```
public void array2() {
    try {
        FileWriter outputStream = new FileWriter("out.txt");
        PrintWriter outs = new PrintWriter(outputStream);

        for (int j = 0; j < 4; j++) {
            outs.println(box[1][j]);
        }
        outs.close();
    }
    catch (IOException ex) {
        System.out.println("i/o error: " + ex.getMessage());
        ex.printStackTrace();
    }
}
```

(iii) [4 marks] What will the following method print out?

```
public void array3() {  
    for (int k = 0; k < 4; k++) {  
        System.out.println(box[k][k]);  
    }  
}
```

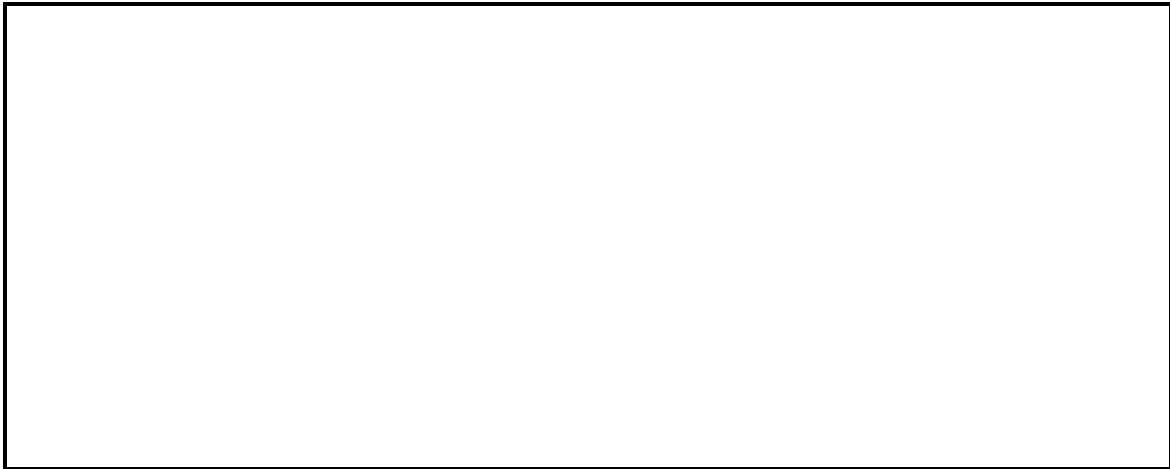
(iv) [7 marks] What will the following method print out?

```
public void array4() {  
    for (int i = 0; i < 4; i++) {  
        for (int j = i+1; j < 4; j++) {  
            System.out.println(box[i][j]);  
        }  
    }  
}
```

(b) [18 marks] **Arrays of Objects** Consider the following program:

```
import javax.swing.JOptionPane;
public class ShopListNew {
    public static void main(String[] args) {
        ShoppingList s = new ShoppingList();
        s.read();
        s.display();
        s.totalCost();
    }
}
class ShoppingList {
    private int MaxNumItems = 40;
    private ShoppingItem[] list;
    private int numItems = 0;
    public ShoppingList() {
        list = new ShoppingItem[MaxNumItems];
    }
    public void read() {
        numItems = 0;
        ShoppingItem e = new ShoppingItem();
        while ((numItems < MaxNumItems) && (e.enter())) {
            list[numItems] = e;
            numItems++;
            e = new ShoppingItem();
        }
    }
    public void display() { //Write your code in part (i)
    }
    public void totalCost() { //Write your code in part (ii)
    }
}
class ShoppingItem {
    private String name;
    private double price;
    private int amount;
    public boolean enter() {
        name = JOptionPane.showInputDialog("Enter Name");
        if (name == null)
            return false;
        String sPrice = JOptionPane.showInputDialog("Enter price");
        price = Double.parseDouble(sPrice);
        String sAmount = JOptionPane.showInputDialog("Enter amount");
        amount = Integer.parseInt(sAmount);
        return true;
    }
    public void print() {
        System.out.println("Name: " + name);
        System.out.println("Price: " + price);
        System.out.println("Amount: " + amount);
    }
    public String getName() {
        return name;
    }
    public double getPrice() {
        return price;
    }
    public double getAmount() {
        return amount;
    }
}
```

(i) [9 marks] Write the `display` method so that it prints out the details (name, price, amount) of each item on the shopping list.



(ii) [9 marks] Write the `totalCost` method so that it calculates and prints out the total cost of all items. The cost for each item is obtained by multiplying its price and amount.



**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. Recursion**

[15 marks]

Consider the following method:

```
public int myMethod(int n) {  
    System.out.println(n);  
    if (n <= 3)  
        return n;  
    else {  
        int f1 = myMethod(n - 2);  
        int f2 = myMethod(n - 1);  
        return f1 - f2;  
    }  
}
```

**(a)** [3 marks] How many recursive steps are there in this method? ( A recursive step solves a problem that is a smaller version of the original problem)

**(b)** [12 marks] What will this method print out if it is called with an argument of 6?

**Question 5. Containers and Genericity**

[30 marks]

Consider this GenericMap interface:

```
public interface GenericMap{
    public boolean insert(Object key, Object value);
    public Object find(Object key);
    public Object remove(Object key);
    public int sizeOf();
}
```

The class GMap implements the GenericMap interface. It is used and performs in the same way as the example from lectures, however, the implementation is a little different. Rather than using an array to store keys, and a separate array to store values, GMap only uses **one** array to store both the keys and the values. In this scheme, if the key is stored at position  $i$ , then the value is stored at position  $i + 1$ .

For simplicity you may assume that all keys are unique, that is, no duplicate keys are passed to the GMap class.

The following code is the beginning of the implementation of the GMap class that implements the GenericMap interface. The constructor has a parameter that specifies the size of the map (the number of key and value pairs) and it creates the array in which the key and value associations are stored.

```
class GMap implements GenericMap{
    private Object data[];
    private int size;

    public GMap(int capacity){
        size = 2 * capacity;
        data = new Object[size];
    }

    // Your code from questions b,c,d and e will fit in here.

    public void printAll(){
        for(int i = 0; i < size; i++)
            System.out.println(data[i]);
    }
}
```

For example, if we were to execute the following code:

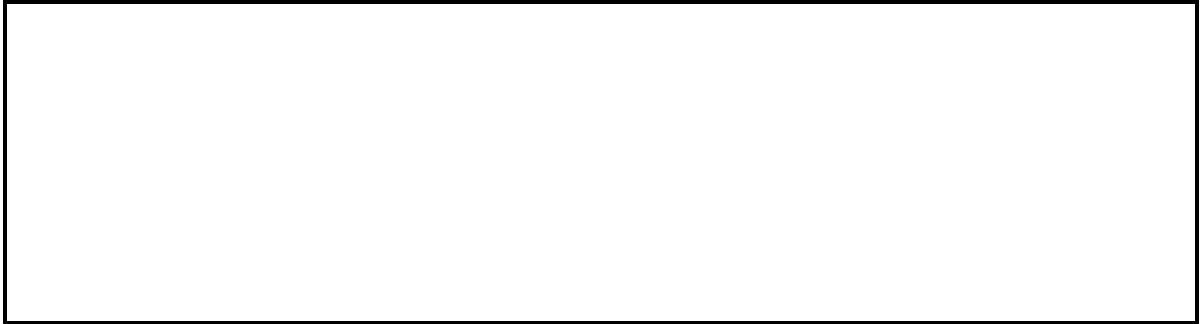
```
GMap map = new GMap(2);
map.insert("bruce", "bruce@email.com");
map.insert("john", "john@email.com");
map.printAll();
```

We would get the following output:

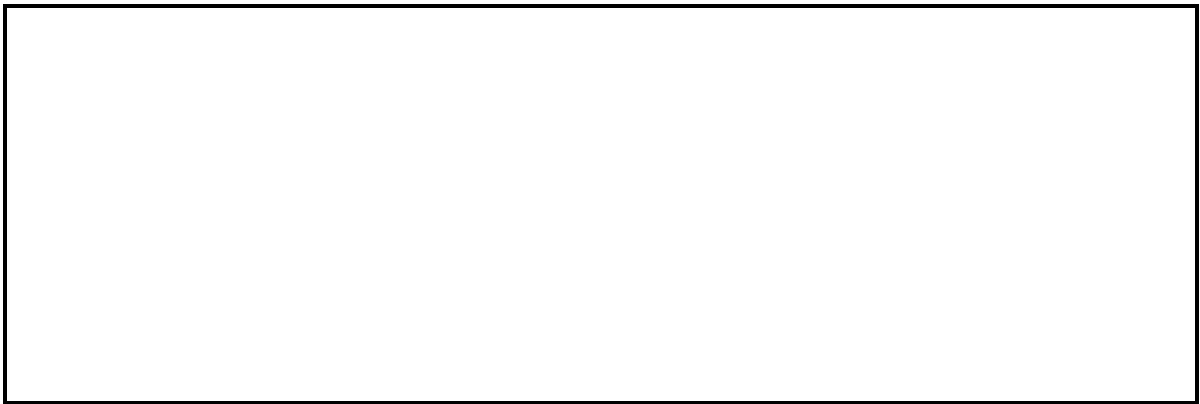
```
bruce
bruce@email.com
john
john@email.com
```

(a) [6 marks] Draw a diagram showing what is in the GMap array *data*, after the following code has been executed. Note: The `remove` method finds a specified key in the GMap and removes the key, value pair.

```
GMap map = new GMap(2);
map.insert("bruce", "bruce@email.com");
map.insert("john", "john@email.com");
map.remove("bruce");
map.find("john");
```



(b) [6 marks] Write the implementation for the `sizeOf` method in the GMap class, as specified by the GenericMap interface. The `sizeOf` method returns the number of key, value pairs contained within the GMap object.



(c) [6 marks] Write the implementation for the `insert` method in the GMap class, as specified by the GenericMap interface. The `insert` method places key and value into the Map and returns true if the operation succeeded. If the insert fails, then the method returns false.



(d) [6 marks] Write the implementation for the `find` method in the `GMap` class, as specified by the `GenericMap` interface. The `find` method takes a specified `key` and returns the corresponding `value`. If the `key` is not found, then the method returns `null`.

(e) [6 marks] Write a small method to test the `GMap` class. The test method is required to insert a name and email address and then print out the result of finding that name.

```
public void test(){
```

**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. Inheritance, Events and Swing**

[30 marks]

Consider the following `TestButton` class. This class is similar to that shown in lectures, but has an additional method called `makeButton()`.

```
import javax.swing.*;
import java.awt.event.*;

class TestButton implements ActionListener{
    private int clicks = 0;
    private JFrame frame;
    private JPanel panel;
    private JLabel label;
    private JButton button;

    public TestButton(){
        frame = new JFrame("Testing a Clicky Button");
        panel = new JPanel();
        label = new JLabel("Number of clicks:  " + String.valueOf(clicks));
        button = makeButton("Button",this);

        panel.add(button);
        panel.add(label);
        frame.getContentPane().add(panel);

        frame.setSize(240,100);
        frame.setVisible(true);
    }

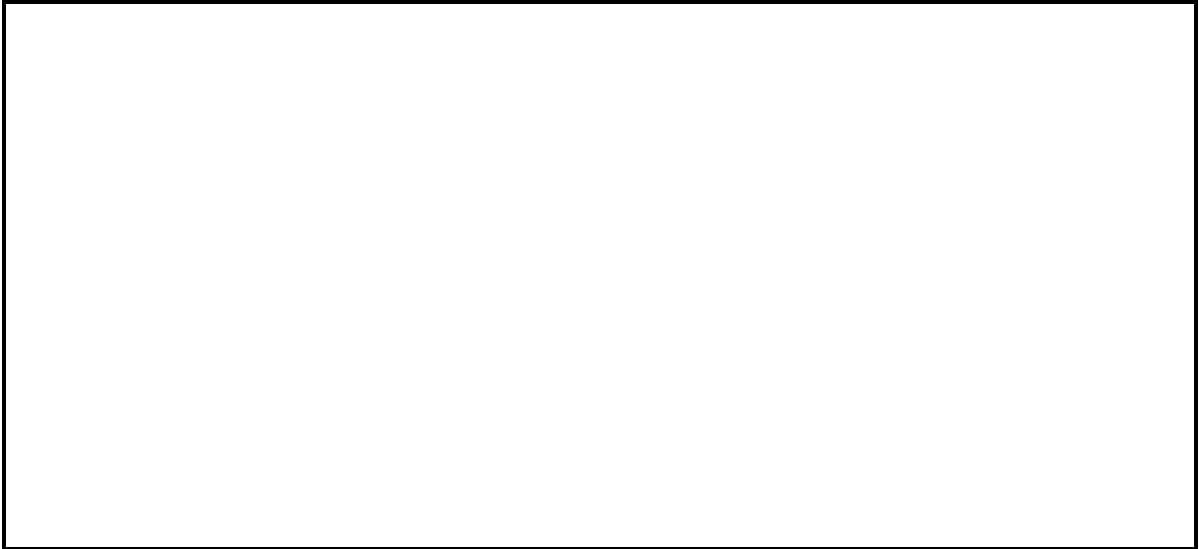
    public JButton makeButton(String title, ActionListener listener){
        JButton button = new JButton(title);
        button.addActionListener(listener);
        return button;
    }

    public void actionPerformed(ActionEvent e){
        label.setText("Number of clicks:  " + String.valueOf(++clicks));
    }
}
}
```

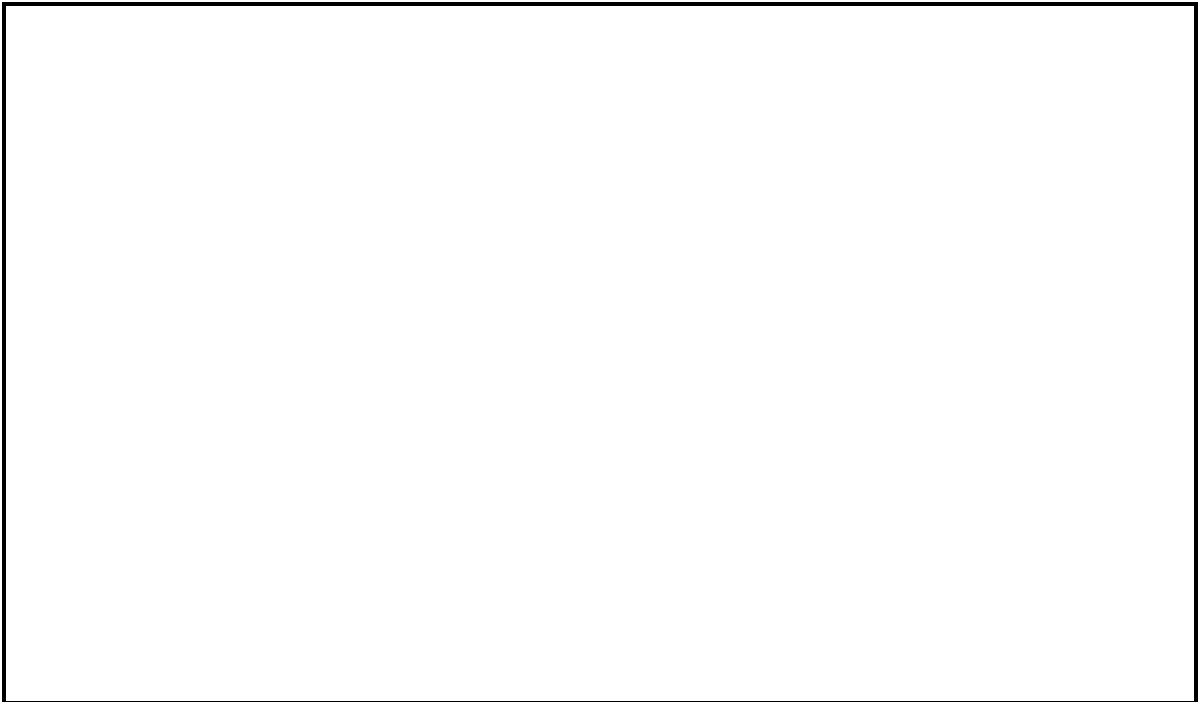
(a) [5 marks] Describe what the `makeButton()` method does.

(b) [4 marks] For what reasons does `TestButton` implement the `ActionListener` interface.

(c) [5 marks] Draw a diagram illustrating the principles of the Java Event model. Clearly indicate how the different parts of `TestButton` fit into your diagram.



(d) [10 marks] Suppose that, rather than using a method like `makeButton()`, we want to achieve the same effect by *subclassing* `JButton` in a new class called `MyButton`. Write the `MyButton` code.



(e) [6 marks] Indicate on the following code (which is the same as code as at the start of this question) **ALL** the changes that are required to use the `MyButton` class that you completed in part (d). Be sure to cross out any unnecessary code.

```
import javax.swing.*;
import java.awt.event.*;

class TestButton implements ActionListener{
    private int clicks = 0;
    private JFrame frame;
    private JPanel panel;
    private JLabel label;
    private JButton button;

    public TestButton(){
        frame = new JFrame("Testing a Clicky Button");
        panel = new JPanel();
        label = new JLabel("Number of clicks:  " + String.valueOf(clicks));
        button = makeButton("Button",this);

        panel.add(button);
        panel.add(label);
        frame.getContentPane().add(panel);

        frame.setSize(240,100);
        frame.setVisible(true);
    }

    public JButton makeButton(String title, ActionListener listener){
        JButton button = new JButton(title);
        button.addActionListener(listener);
        return button;
    }

    public void actionPerformed(ActionEvent e){
        label.setText("Number of clicks:  " + String.valueOf(++clicks));
    }
}
```

\*\*\*\*\*

**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.