**EXAMINATIONS — 2004**

END-OF-YEAR

**COMP 102**
**INTRODUCTION TO**
**COMPUTER PROGRAM DESIGN**
**Unchecked Answers**

**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.

There are spare pages (pages 7, 8, 10, 14, 24) for your working and your answers in this exam.

## Questions

| | **Marks** |
|---|---|
| 1. Understanding Java programs | [50] |
| 2. Writing simple Java Programs | [30] |
| 3. Classes and objects | [25] |
| 4. Files and collections | [25] |
| 5. Inheritance | [20] |
| 6. GUI | [15] |
| 7. Recursion | [15] |

## Question 1. Understanding Java programs [50 marks]

In parts **(a)** to **(g)**, you should assume each method is declared in a suitable class and called on an object of that class. The name and other details of this class are unimportant as the methods do not refer to fields or other methods of the class.

**(a)** [3 marks] What will be printed when the following method `q1a` is called?

```
public void q1a() {
  int x = 3;
  int y = 10;
  int z = x*y;
  y = z-y;
  x = x+y;
  z = y/3;
  System.out.println(x);
  System.out.println(y);
  System.out.println(z);
}
```

23202

**(b)** [7 marks] What will be printed when the following method `q1b` is called?

```
public void q1b() {
  String s = "Victoria";
  System.out.println( s.length() );
  System.out.println( s.charAt(5) );
  int a = s.indexOf('i');
  System.out.println( a );
  int b = s.indexOf('i', a+1);
  System.out.println( b );
  System.out.println( s.substring(0, 4) );
  System.out.println( s.substring(a, b) );
  System.out.println( s.substring(b));
}
```

8r16Victictoria

**(c)** [6 marks] Consider the following method definition:

```
public void q1c(int a, int b) {
  int c, d;
  c = a-b;
  if ( a < b)
    c = b-a;
  if ( a == c )
    d = a;
  else
    d = b;
  System.out.println(c + " " + d);
}
```

**(i)** [2 marks] What will be printed when method `q1c` is called with **1** and **5** as its arguments?

4 5

**(ii)** [2 marks] What will be printed when method `q1c` is called with **5** and **1** as its arguments?

4 1

**(iii)** [2 marks] What will be printed when method `q1c` is called with **3** and **3** as its arguments?

0 3

**(d)** [6 marks] What will be printed when the following method `q1d` is called?

```
public void q1d() {
  int s = 0;
  int k = 0;
  while ( s < 30 ) {
    System.out.print( k + " " + s);
    s = s + 3*k + 1;
    k = k + 1;
  }
  System.out.println(s-k);
}
```

0 01 12 53 124 2230

**(e)** [6 marks] What will be printed when the following method `q1e` is called?

```
public void q1e() {
  int[] a = {4, 8, 3, 0, -5, 7, 2, 9, 3};
  int s = 0;
  for (int i = 0; i < a.length; i=i+2) {
    System.out.println(i + " " + a[i]);
    s = s + a[i];
  }
  System.out.println(s);
}
```

0 42 34 -56 28 37

**(f)** [6 marks] What will be printed when the following method `q1f` is called?

```
public void q1f() {
  int[] a = new int[6];
  int[] b = new int[10];
  for (int k = 0; k < a.length; k++) {
    a[k] = k;
  }
  for (int k = 0; k < a.length; k++) {
    b[k] = 2*a[k];
  }
  for (int k = 0; k < a.length; k++) {
      System.out.println(a[k] + " " + b[k]);
  }
}
```

0 01 22 43 64 85 10

**(g)** [8 marks] What will be printed when the following method `q1g` is called?

```
public void q1g() {
  int m = 4;
  int[][] a = new int[m][m];
  for (int i = 0; i < m; i++) {
    for (int j = 0; j < m; j++) {
      a[i][j] = i*m+j;
      System.out.print(a[i][j] + " ");
    }
    System.out.println();
  }
  for (int i = 0; i < m; i++) {
    System.out.println(a[i][i] + " " + a[i][m-i-1]);
  }
}
```

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 3 5 6 10 9 15 12

**(h)** [8 marks] What will the following program print out?

```
public class Q1h {
  public static void main(String[] args) {
    Car a = new Car("Ford");
    Car b = new Car("Fiat");
    Car w = null;
    a.moveCar(10);
    b.moveCar(8);
    b.moveCar(8);
    a.moveCar(10);
    if ( a.getPos() > b.getPos() )
      w = a;
    else if ( a.getPos() < b.getPos() )
      w = b;
    if ( w == null )
      System.out.println("Race is drawn");
    else
      System.out.println(w.getName() + " wins by " +
                         Math.abs(a.getPos()-b.getPos()));
  }
}

class Car {
  private String name;
  private int pos = 0;
  public Car(String n) {
    name = n;
  }
  public void moveCar(int x) {
    pos = pos + x;
    System.out.println(name + " moves to " + pos);
  }
  public int getPos() {
    return pos;
  }
  public String getName() {
    return name;
  }
}
```

Ford moves to 10 Fiat moves to 8 Fiat moves to 16 Ford moves to 20 Ford wins by 4

## 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.

## 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 simple Java programs [30 marks]

**(a)** [15 marks] **Strings and conditionals**

You are to write a program that reads a string, consisting of two sequences of characters separated by a space, then prints a message showing:

- the input string,
- the length of the input string,
- the number of characters before the space, and
- the number of characters after the space.

If the input string does not contain a space, or if it contains more than one space, the program should print the input string and its length, and an error message indicating that why the input is invalid.

For example, the following tables shows the output that should be produced for three possible inputs:

| Input | Output |
|---|---|
| `123 abcd` | `Input: 123 abcd`<br>`Length: 8`<br>`Before: 3`<br>`After: 4` |
| `a+b-c+d*e` | `Input: a+b-c+d*e`<br>`Length: 9`<br>`Error: no space` |
| `f(a, bb, ccc)` | `Input: f(a, bb, ccc)`<br>`Length: 13`<br>`Error: too many spaces` |

Complete the following program so that it behaves as described above. You may print output using either `System.out` or `JOptionPane` methods.

```
import javax.swing.*;
public class StringReader {
  public static void main(String[] args) {
    // Read and analyse a string









  }
}
```

## 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] **Loops and arrays**

You are to write a method that takes two arguments: an array of names (strings) and an array of marks (integers), and prints the data in a table. Each line of the table consists of a name from the first array and the corresponding mark from the second array, separated by a space.

For example, if the method was called as follows:

```
String[] n = {"Alan", "Lindsay", "Sharon"};
int[] m = {50, 30, 80};
printTable(n, m);
```

the output would be:

```
Alan 50
Lindsay 30
Sharon 80
```

Complete the following method so that it behaves as described above. If the arrays are not of the same length, the method should print a message saying "`Invalid data`". You may print output using either `System.out` or `JOptionPane` methods.

```
public void  printTable(String[] names, int[] marks) {









}
```

**Question 3. Classes and Objects** [25 marks]

The following is an outline of a class for composing and formatting messages to be displayed by a program:

```
pubic class Message {

  private String msg = "";

  // Create an empty message
  public Message() {
  }

  // Add a string to the message, along with a space to separate
  // any subsequent string
  public void append(String s) {
    msg = msg + s + " ";
  }

  // Continue the message on a new line
  public void nl() {
    msg = msg + "\n";
  }

  // Return the current message
  public String getMessage() {
    return msg;
  }

  // Clear the current message
  public void clear() {
    msg = "";
  }

}
```

**(a)** [10 marks]

What output will be produced by the following program, assuming that `Message` is defined as above?

```
public class Messenger {
  public static void main(String[] args) {
    Message m = new Message();
    m.append("I");
    m.append("saw");
    m.append("a");
    m.append("shining");
    m.append("light.");
    System.out.println( m.getMessage() );
    m.clear();
    m.append("It");
    m.append("was");
    m.append("very,");
    m.nl();
    m.append("very");
    m.append("bright.");
    System.out.println( m.getMessage() );
  }
}
```

## 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]

You are now required to modify `Message`, so that the constructor is passed an integer giving the maximum number of characters that may appear on any line of a message. The `append` method must now check whether the string to be added will fit on the current line, and if not it should start a new line. The easiest way to do this is to add a field to record the number of characters on the current line, and update it whenever the current line changes.

Show the modifications necessary on the following copy of `Message`, to make `Message` behave as described above:

```
public class Message {
  private String msg = "";



  public Message(                                  ) {



  }

  public void append(String s) {









    msg = msg + s + " ";




  }

  public void nl() {

    msg = msg + "\n";

  }

  public String getMessage() {

    return msg;
  }

  public void clear() {

    msg = "";

  }
}
```

## Question 4. Files and collections [25 marks]

Suppose we have a file containing data about products, saved in the following format:

code
price

Assume the file `in.txt` contains the following data for three products.

```
1555
3.4
2666
4.5
1555
7.8
```

**Answer part (a) and (b) using the example file given above.**
**The input and output files in these subquestions do not raise I/O exceptions.**

**(a)** [15 marks] Consider the following `Product` class:

```
import java.io.*;

class Product {
  private String code;
  private double price;

  public Product(String c, String p) {
    code = c;
    price = Double.parseDouble(p);
  }

  public Product() {
  }

  public void print() {
    System.out.println("Code: " + code);
    System.out.println("Price: " + price);
  }

  public void load(BufferedReader f)
         throws NumberFormatException, IOException {
    code = f.readLine();
    String sPrice = f.readLine().trim();
    price = Double.parseDouble(sPrice);
  }

  public void save(PrintWriter f) {
    f.println(code);
    f.println(price);
  }

  public boolean equals(Object a) {
        Product aa = (Product) a;
        return this.code.equals(aa.code);
  }

}
```

What will the following program print out to the screen?

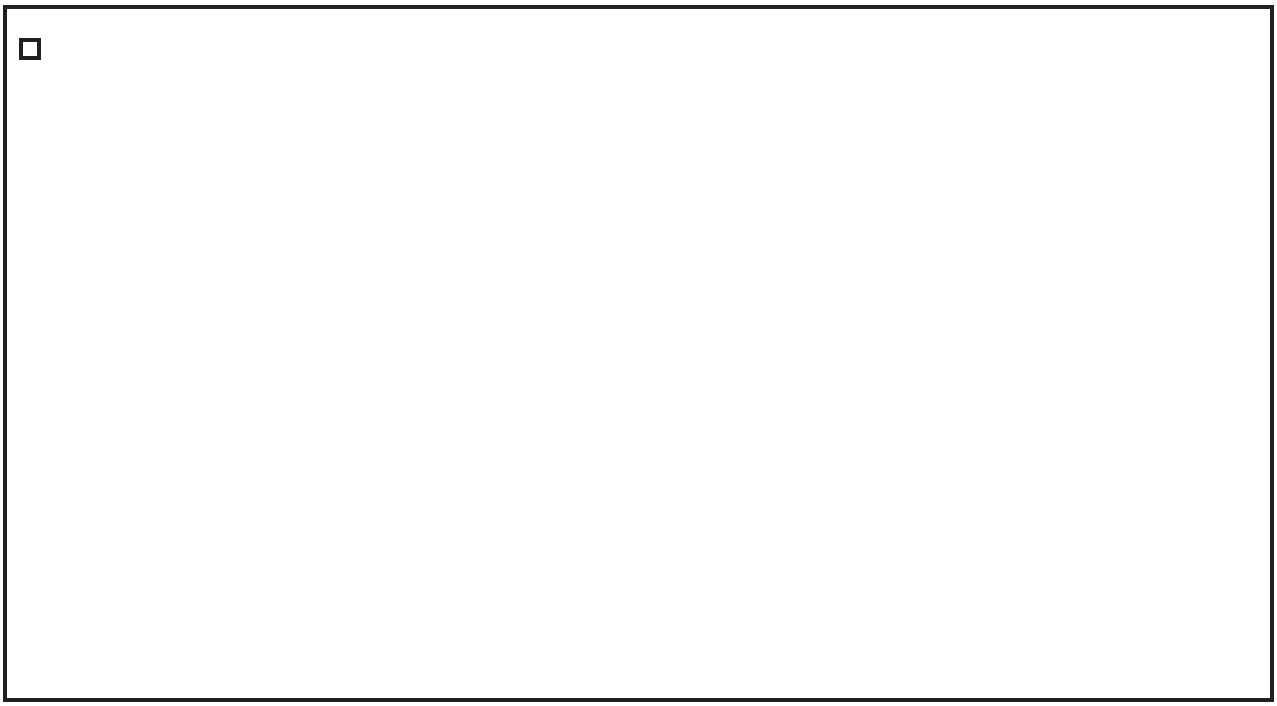
```
import java.io.*;

public class Files1 {
  public static void main(String[] args) {
    try {
      FileReader inStream = new FileReader("in.txt");
      BufferedReader ins = new BufferedReader(inStream);
      String d1 = ins.readLine();
      String d2 = ins.readLine();
      System.out.println(d1 + " " + d2);

      Product a, b;
      System.out.println("a:");
      a = new Product(d1,d2);
      a.print();

      System.out.println("b:");
      b = new Product();
      b.load(ins);
      b.print();

      ins.close();
    }
    catch (NumberFormatException ex) {
      System.out.println("Wrong Number format!");
    }
    catch (IOException ex) {
      System.out.println("File I/O Error");
    }
  }
}
```



**(b)** [10 marks] Consider the following `Set` class:

```
class Set {
  private Object elements[];
  private int MAX = 500;
  private int pos;

  public Set() {
    elements = new Object[MAX];
  }

  public boolean addElement(Object x) {
    if (containsElement(x))
      return false;
    for (int i=0; i < MAX; i++) {
      if (elements[i] == null) {
        elements[i] = x;
        return true;
      }
    }
    return false;
  }

  public boolean containsElement(Object x) {
    for (int i=0; i < MAX; i++) {
      if (elements[i] != null
          && elements[i].equals(x)) {
        return true;
      }
    }
    return false;
  }

  public Object findElement(Object x) {
    for (int i = 0; i < MAX; i++) {
      if (elements[i] != null
               && elements[i].equals(x)) {

          return elements[i];
      }
    }
    return null;
  }

  public void getElementInit() {
    pos = 0;
  }

  public Object getElement() {
    while (pos < MAX) {
      if (elements[pos] != null) {
        return elements[pos++];
      }
      pos++;
    }
    return null;
  }
}
```

What will the following program print out to the screen?

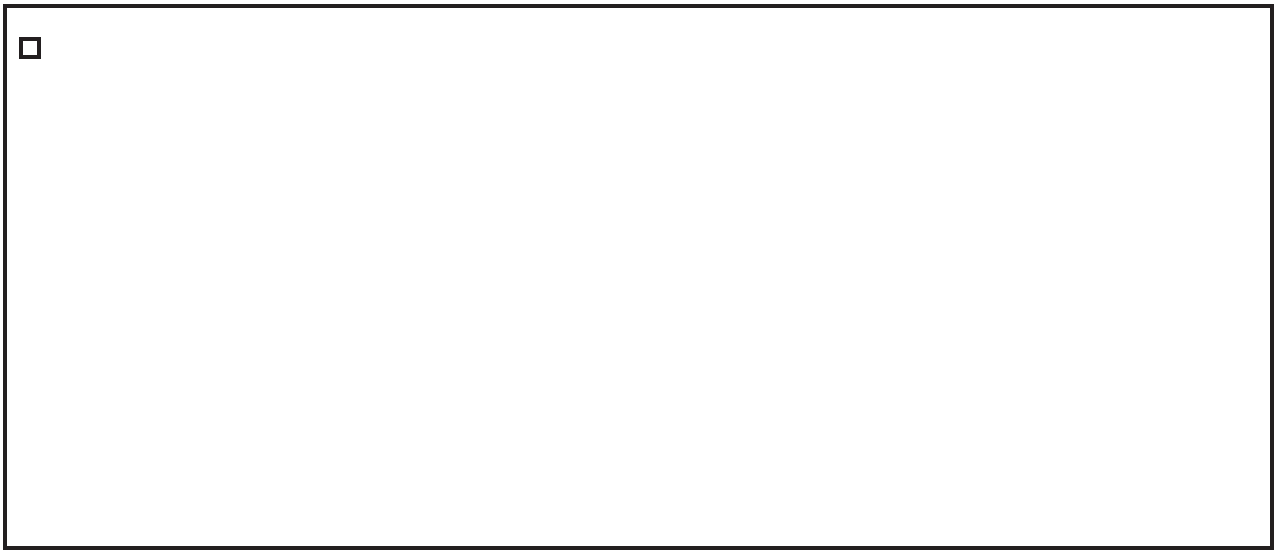
```
import java.io.*;

public class Files2 {
  public static void main(String[] args) {
      Set s = new Set();

      Product p = new Product("34", "2.5");

      boolean r = s.addElement(p);
      System.out.println(r);

      Product b = (Product) s.findElement(p);
      b.print();
  }
}
```



**Question 5. Inheritance** [20 marks]

Consider the following code:

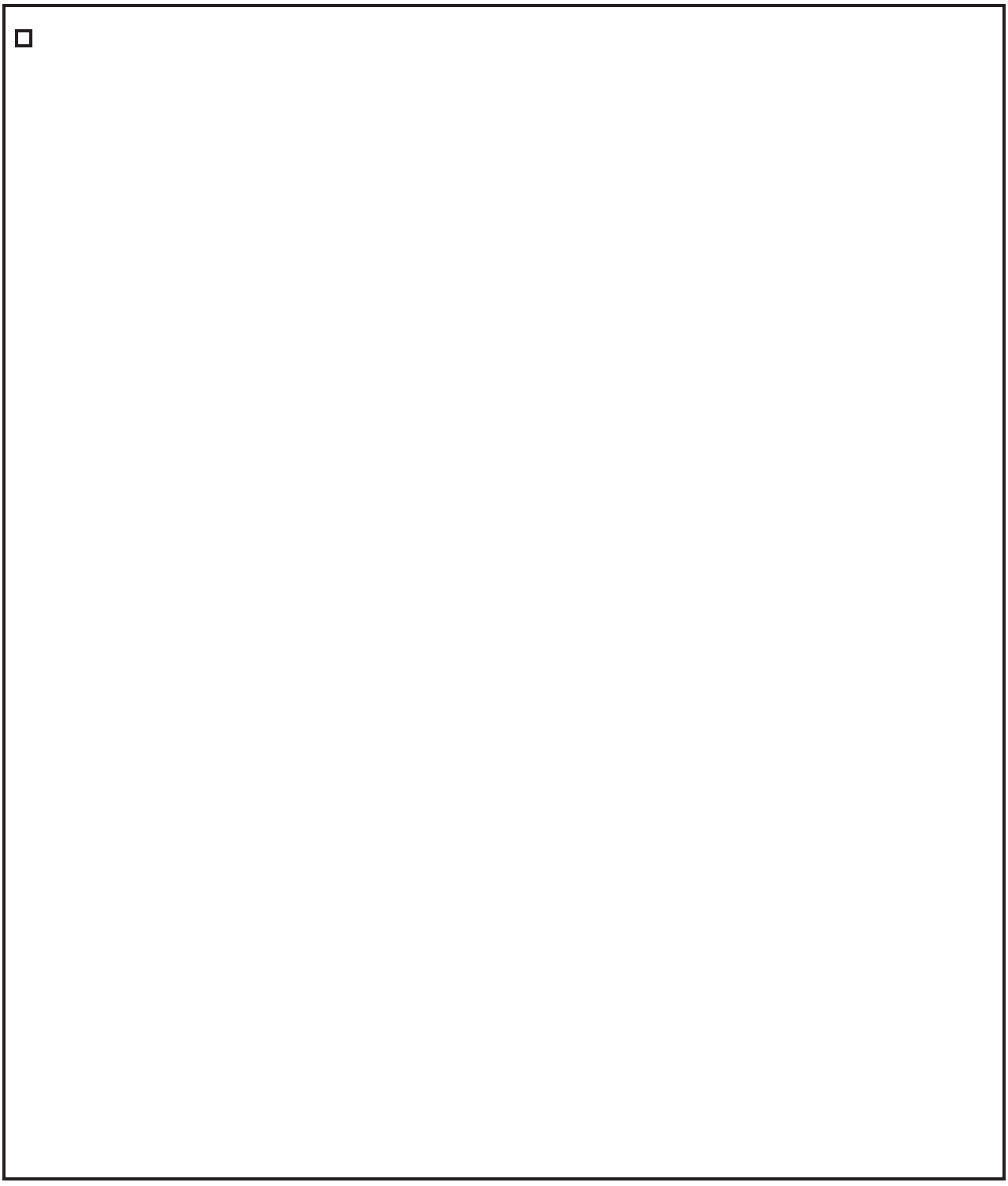
```
import javax,swing,*;
class Book {
    private String code;
    private String author;
    private boolean in = true;
    private String borrowerId = null;
    public void enter() {
        code = JOptionPane.showInputDialog("Enter code: ");
        author = JOptionPane.showInputDialog("Enter author: ");
    }
    public void returnItem() {
        in = true;
        borrowerId = null;
    }
    public void borrowItem(String id) {
        in = false;
        borrowerId = id;
        System.out.println("   Four weeks loan");
    }
    public String getCode() {
        return code;
    }
    public String getAuthor() {
        return author;
    }
    public boolean getStatus() {
        return in;
    }
    public String getBorrowerID() {
        return borrowerId;
    }
    public void setCode(String c) {
        code = c;
    }
    public void setAuthor(String a) {
        author = a;
    }
    public void setStatus(boolean b) {
        in = b;
    }
    public void setBorrowerID(String d) {
        borrowerId = d;
    }
    public String toString() {
        return "Book  "+ code + "  " + author;
    }
}
```

Write a `TextBook` class to extend the `Book` class. The differences between the `TextBook` class and the `Book` class are as follows:

- `TextBook` class has an extra data field `courseName`.
- When the `enter()` method is called, it should ask for the code, the author and the course name.
- The loan period for textbooks is "three days" instead of "four weeks". Make sure the `borrowItem` method prints the right message.

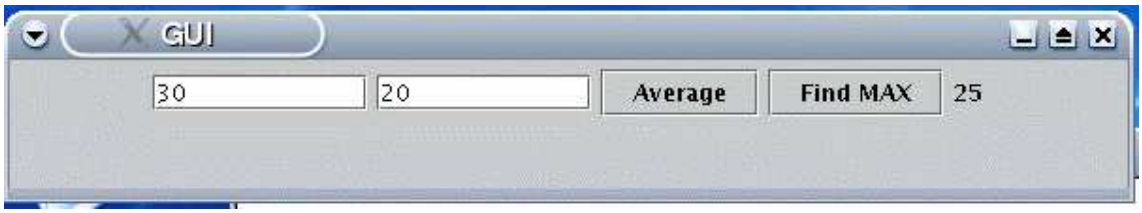
- When the `toString()` method is called, it should return a string starting with “Text book”, followed by the code, the author, and the course name.

**Note you should NOT change the `Book` class and you should NOT redeclare the private data fields of the `Book` class in your `TextBook` class.**



**Question 6. GUI** [15 marks]

You are required to write a program that creates a simple GUI with two text fields, two buttons and one label as shown in the figure below.




Your program should allow users to give two numbers in the two text fields.The label should show the average of the two numbers if the `Average` button is clicked, or show the maximum of the two numbers if the `Find MAX` button is clicked.

Some code has been written for you and your task is to add two buttons to the program and make sure that the program responds to the button clicks. You do not need to worry about the layout.

Show your answers by modifying/completing the code given on next page.

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

public class NumberGUI {
  public static void main (String[] args) {
    SimpleGUI n = new SimpleGUI();
  }
}
class SimpleGUI implements ActionListener {
  private JFrame frame;
  private JPanel panel;




  private JLabel label;
  private JTextField text1;
  private JTextField text2;

  public SimpleGUI() {
    frame = new JFrame("GUI");
    panel = new JPanel();




    text1 = new JTextField(10);
    text2 = new JTextField(10);
    label = new JLabel();
    panel.add(text1);
    panel.add(text2);




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


    frame.setSize(600, 100);
    frame.setVisible(true);
  }
}
```

## 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 7. Recursion [15 marks]

Consider the following program:

```
import javax.swing.JOptionPane;

public class RecursionProgram {
  public static void main(String[] args) {
    Recursion p = new Recursion();
    int result = p.myMethod(10, 7);
    System.out.println("my method: " + result);
  }
}

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

**(a)** [3 marks] What are the stopping cases for the recursive method?

**(b)** [12 marks] What will the program print out?

Note: Each time the recursive method is called, one of the arguments is printed. The program also prints the final result.

**********

## 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.