TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI

VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS — 2005

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 marks.

Non-programmable calculators without a full alphabetic key pad are permitted.

Non-electronic foreign language dictionaries are permitted.

There are spare pages for your working and your answers in this exam.

Questions

	Marks
1. Understanding Java programs	[45]
2. Writing simple Java Programs	[30]
3. Classes and objects	[15]
4. Files	[15]
5. Composition	[20]
5. Inheritance	[20]
6. GUI	[15]
7. Recursion	[20]

Question 1. Understanding Java programs

[45 marks]

In parts **(a)** to **(g)**, you should assume each method is declared in a suitable class and called on an object of this 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 = 5;
  int y = 4;
  int z = x*y;
  y = z-y;
  System.out.println(x);
  System.out.println(y);
  System.out.println(z);
}
```

5
16
20

(b) [5 marks] What will be printed when the following method `Q1b` is called?

```
public void Q1b() {
  String s = "Wellington";
  System.out.println( s.length() );
  System.out.println( s.charAt(1) );
  int a = s.indexOf('i');
  System.out.println( a );
  System.out.println( s.substring(0, a) );
  System.out.println( s.substring(a+1));
}
```

10
e
6
Well
ton

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

```
public void Q1c(int a, int b) {
  int c = 0;
  int d;
  if ( a < b) {
    c = a+b;
  }
  d = 2*b;
  if ( a > c )
    c = a;
  else
    d = b;
  System.out.println(c + " " + d);
}
```

(i) [2 marks] What will be printed when method `Q1c` is called with **3** and **0** as its arguments?

3 0

(ii) [2 marks] What will be printed when method `Q1c` is called with **2** and **2** as its arguments?

2 4

(iii) [2 marks] What will be printed when method `Q1c` is called with **5** and **1** as its arguments?

5 2

(d) [5 marks] What will be printed when the following method `Q1d` is called?

```
public void Q1d() {
  int s = 25;
  int k = 1;
  while ( s > 0 ) {
    System.out.println( k + " " + s);
    s = s - 2*k + 1;
    k = k + 1;
  }
}
```

1 25
2 24
3 21
4 16
5 9

(e) [4 marks] What will be printed when the following method `Q1e` is called?

```
public void Q1e() {
  int[] a = {2, 0, -1, 4};
  int s = 0;
  for (int i = 0; i < a.length; i++) {
    System.out.println(a[i] + " " + s);
    s = s + a[i];
  }
}
```

2 0
0 2
-1 2
4 1

(f) [6 marks] What will be printed when the following method `Q1f` is called?

```
public void Q1f() {
  int m = 3;
  int[] a = new int[2*m];
  for (int k = 0; k < a.length; k++) {
    a[k] = 3*k;
    System.out.print(a[k] + " ");
  }
  System.out.println();
  for (int k = 0; k < m; k++) {
    System.out.println(a[2*k] + a[a.length-k-1]);
  }
}
```

0 3 6 9 12 15
15
18
21

(g) [9 marks] What will be printed when the following method `Q1g` is called?

```
public void Q1g() {
  int m = 3;
  int k = 1;
  int[][] a = new int[m][m];
  for (int i = 0; i < m; i++) {
    for (int j = 0; j < m; j++) {
      a[i][j] = k;
      k = k+1;
      System.out.print(a[i][j] + " ");
    }
    System.out.println();
  }
  for (int i = 0; i < m; i++) {
    System.out.print(a[i][m-i-1] + " ");
  }
  System.out.println();
}
```

1 2 3
4 5 6
7 8 9
3 5 7

(h) [7 marks] What will the following program print out?

```
public class Q1h {
  public static void main(String[] args) {
    int[] points1 = {5, 8, 10};
    int[] points2 = {9, 8, 7};
    Contestant a = new Contestant("Chris");
    Contestant b = new Contestant("Susan");
    for (int k = 0; k < points1.length; k++) {
      a.addScore(points1[k]);
      b.addScore(points2[k]);
      System.out.println(
        a.getName() + " has " + a.getScore() + " " +
        b.getName() + " has " + b.getScore());
    }
    String result = "Results is a tie";
    if ( a.getScore() > b.getScore() )
      result = a.getName() + " wins";
    else if ( a.getScore() < b.getScore() )
      result = b.getName() + " wins";
    System.out.println(result);
  }
}

class Contestant {
  private String name;
  private int score = 0;
  public Contestant(String n) {
    name = n;
  }
  public void addScore(int x) {
    score = score + x;
  }
  public int getScore() {
    return score;
  }
  public String getName() {
    return name;
  }
}
```

Chris has 5 Susan has 9
Chris has 13 Susan has 17
Chris has 23 Susan has 24
Susan wins

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 and analyses it. If the string contains exactly one space, the program should print the string with the space replaced by a colon (`:`). If the string contains no space or more than one space, the program should print `Invalid input`.

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

Input	Output
`123 abcd`	`123:abcd`
`a+b-c+d*e`	`Invalid input`
`f(a, bb, ccc)`	`Invalid input`

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 StringAnalyser {
  public static void main(String[] args) {
    // Read and analyse a string
    String s = JOptionPane.showInputDialog("Enter string");

   int p = s.indexOf(" ");
   if (p == -1) {
        System.out.println("No space");
    } else {
        int q = s.indexOf(" ", p+1);
        if (q != -1) {
           System.out.println("Too many");
        } else {
           System.out.println(p);
           String m = s.substring(0, p) + ":";
           m = m + s.substring(p+2, s.length()-1);
           System.out.println(m);
        }
    }


  }
}
```

(b) [15 marks] **Loops and arrays**

You are to write a method that takes as arguments two arrays of integers and prints a table with three columns, as follows. The first column should show the values in the first array; the second column should show the values of the second array; and the third column should show the difference between the two values on that row. The difference should be displayed with a leading plus sign (+) if the number in the first column is bigger and a minus sign (−) if the number in the second column is bigger. An equals sign (=) should be printed in place of the difference if the two values are equal.

You may assume that the two arrays have the same size, and you do not need to make the columns line up. You may print output using either `System.out` or `JOptionPane` methods.

For example, if the method was called as follows:

```
int[] avals = {17, -5, 18, 15, 12};
int[] bvals = {11, 12, 27, 15, -9};
printTable(avals, bvals);
```

the output would be:

```
17 11 +6
-5 12 -17
18 27 -9
15 15 =
12 -9 +21
```

Complete the following method so that it behaves as described above.

```
public void  printTable(int[] a, int[] b) {

   for (int i = 0; i ¡ a.length; i++) {

      String m = a[i] + " " + b[i] + " ";

      if ( a[i] ¡ b[i] )
         m = m + (a[i] - b[i]);
      else if ( a[i] ¿ b[i] )
         m = m + "+" + (a[i] - b[i]);
      else m = m + "=";

      System.out.println(m);

   }

}
```

Question 3. Classes and Objects [15 marks]

The following is an outline of a class for printing text arranged in groups of lines, such as verses of a poem or song. The method `printLine` is used to print a line, and `newGroup` is used to start a new group of lines, printing a blank line to separate the new group from the previous group.

```
public class LinePrinter {
  public LinePrinter() {
  }
  public void printLine(String s) {
    System.out.println(s);
  }
  public void newGroup() {
    System.out.println();
  }
}
```

You are to modify this class so that lines are numbered within each group. Also, the `LinePrinter` should take an argument giving the maximum number of lines in each group, and a new group should be started if more than that number of lines are printed since the current group was started.

For example, once you have completed this modification, executing the program shown below on the left should produce the output shown below on the right.

<table>
<tr>
<td>

```
public class Printer {
  public static void main(String[] args) {
    LinePrinter p = new LinePrinter(3);
    p.printLine("One");
    p.printLine("Two");
    p.printLine("Three");
    p.printLine("Four");
    p.printLine("Five");
    p.newGroup();
    p.printLine("Six");
    p.printLine("Seven");
    p.printLine("Eight");
    p.printLine("Nine");
    p.printLine("Ten");
  }
}
```

</td>
<td>

```
1 One
2 Two
3 Three

1 Four
2 Five

1 Six
2 Seven
3 Eight

1 Nine
2 Ten
```

</td>
</tr>
</table>

Show the modifications necessary on the following copy of `LinePrinter`:

```
public class LinePrinter {

    private int pageSize;
    private int line = 0;

    public LinePrinter(   int p                        ) {

       pageSize = p;

    }

    public void printLine(String s) {

      if ( line == pageSize )
         newPage();
      line++;

         System.out.println(s);

    }

    public void newGroup() {

      System.out.println();
      line = 0;

         System.out.println();

    }
}
```

****