TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Student ID: .....

EXAMINATIONS — 2008

END-OF-YEAR

COMP 102
INTRODUCTION TO COMPUTER PROGRAM DESIGN

Time Allowed: 3 Hours ******** WITH SOLUTIONS **********

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 is documentation at the end of the paper, which you may tear off.
This includes example programs showing a variety of Java syntax.

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

Questions

	Marks
1. Basic Java	[63]
2. Files and Arrays of Objects	[22]
3. Arrays of Numbers	[46]
4. Recursion	[17]
5. Debugging and Writing Loops	[32]

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. Basic Java [63 marks]

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

```
double x = 8.4;
int y = 5;
double ans = x + y / 2;
x = x / 4;
y = y / 2;
System.out.println("x=" + x);
System.out.println("y=" + y);
System.out.println("ans=" + ans);
```

```
x=2.1
y=2
ans=10.4
```

(b) [8 marks] Consider the following compute method:

```
public String compute(String s) {
    String ans = "yes";

    if (s.length() > 5){
        if (s.equalsIgnoreCase("Saturday") || s.equalsIgnoreCase("Sunday"))
            ans = "good";
        else
            ans = "bad";
    }
    else
        ans = "no";

    return ans;
}
```

What would the following calls to compute return?

compute("November") ⟹ bad

compute("Saturday") ⟹ good

compute("Six") ⟹ no

compute("Today") ⟹ no

(Question 1 continued on next page)

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

```
for ( int k = 10;  k < 25; k = k + 3 ){
    System.out.println("k = " + k);
}
System.out.println("Done");
```

```
k = 10
k = 13
k = 16
k = 19
k = 22
Done
```

(d) [10 marks] What will the following fragment of Java print out?

```
int n = 2;
int m = 10;
while (n < m){
  for( int i = 0; i < n; i++){
        System.out.print(i + " ");
  }
  n = n + 3;
  System.out.println ();
}
System.out.printf("Final: n = %d, m = %d\n", n, m);
}
```

```
0 1
0 1 2 3 4
0 1 2 3 4 5 6 7
Final:  n = 11, m = 10
```

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)

The Item class on the facing page defines Item objects, which have two fields and two methods. The testObjects class has a main method for testing.

(e) [15 marks] If the main method in the testObjects class is called, what will it print out?

```
A: 8
B: 7
C: 10
D: 5
E: 12
F: 14
G: 16
H: 20
0:  40
1:  12
2:  20
```

(f) [6 marks] Write an additional method called third for the Item class which returns true if the two fields of an Item object satisfy the condition: x is greater than 20 or y is smaller than 10. It should return false in all other cases.

```
public boolean third(){
    return ((this.x > 20) || (this.y < 10));
}
```

(Question 1 continued on next page)

(Question 1 continued)

```
class Item {
  private int x, y;
  public Item(int m, int n) {
    this.x = m;
    this.y = n;
  }
  public int first (int i) {
    this.x = this.x + i;
    return this.x;
  }
  public int second() {
    return 2 * this.y;
  }
}
public class testObjects {
  public static void main(String[] args) {
    Item a = new Item(2, 4);
    Item b = new Item(3, 6);
    Item c = new Item(5, 7);

    System.out.println("A: " + a.second());
    System.out.println("B: " + a.first (5));
    System.out.println("C: " + a.first (3));
    System.out.println("D: " + b.first (2));
    System.out.println("E: " + b.second());
    System.out.println("F: " + c.second());
    System.out.println("G: " + (a.second() + b.first (3)));

    Item[] allItems = new Item[10];
     int count = 0;
     allItems[count] = new Item(10, 20);
    count++;
     allItems[count] = b;
    count++;
     allItems[count] = new Item(5, 10);
    count++;


    System.out.println("H: " + allItems[2].second());

    for(int i = 0; i < count; i++){
        System.out.println(i + ": " + allItems[i].second());
    }
  }
}
```

(g) [12 marks] Complete the following definition of a Flight class. Flight objects should have two fields to store the flight number and the destination (e.g NZ402 Auckland). The class should have a constructor that takes two arguments and initialises the fields using its arguments. The class should have two methods – one returns the flight number and the other returns the destination.

```
public class Flight {
private String flightNo;
private String destination;

public Flight(String n, String d){
  this.flightNo = n;
  this.destination = d;
}

public String getFlightNo() {
  return this.flightNo;
}

public String getDestination() {
  return this.destination;
}




}
```

Student ID: .....

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. Files and Array of Objects [22 marks]

(a) [6 marks] Suppose the file `flightdata.txt` contains data about the daily departures of six international flights, where each line contains the flight number and the destination:

```
QF521  Sydney
UA314  Boston
BA476  London
ZA932  Harare
QF145  London
SA647  Durban
```

What will testFiles("`flightdata.txt`") print to System.out?

```
public void testFiles(String fname) {
    try{
        Scanner fileScan = new Scanner(new File(fname));
        int i = 0;
        while (fileScan.hasNext()){
            String word = fileScan.next();
            String line = fileScan.nextLine();
            System.out.println(i + "  " + word);
            i++;
        }
        fileScan.close();
    }
    catch(IOException e){System.out.println("File reading failed: "+e);}
}
```

```
0 QF521
1 UA314
2 BA476
3 ZA932
4 QF145
5 SA647
```

(Question 2 continued on next page)

(Question 2 continued)

(b) [16 marks] Complete the following readStorePrintData method so it reads data from a file in the format described in question 2(a), stores the data in an array of Flight objects, and prints all flight information out using data in the array.

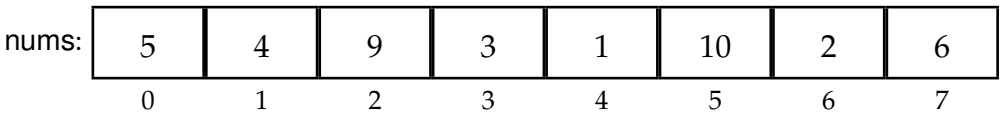
You should use an array of objects and a count to store the flight data. You are required to use the Flight class described in question 1(g). Please note that you should try to complete this question even if you didn't do question 1(g).

```
public void readStorePrintData(String fname){
    try{ Flight [] flights = new Flight[100];
         int count = 0;
         Scanner fileScan = new Scanner (new File (fname));
         while (fileScan.hasNext()&& count < flights.length){
              String no = fileScan.next();
              String des = fileScan.next();
              flights [count] = new Flight(no, des);
              count++;
         }
         for( int i = 0; i < count; i++){
           System.out.println( flights [i].getFlightNo() + "  " + flights [i].getDestination());
         }
         fileScan.close();
    }
    catch(IOException e){System.out.println("File reading failed: "+e);}
}
```

Question 3. Arrays of numbers [46 marks]

(a) [8 marks] Suppose the variable nums is declared to be an array containing 8 integers:

```
int [ ] nums = {5, 4, 9, 3, 1, 10, 2, 6};
```



What will the following code fragment print out?

```
System.out.println(nums[5]);
System.out.println(nums[1] + nums[4−1]);
System.out.println(nums[nums[4]+2]);

int index=0;
while( index < nums.length ){
  System.out.print(index + ": " + nums[index]+ ", ");
  index = index + 3;
}
```

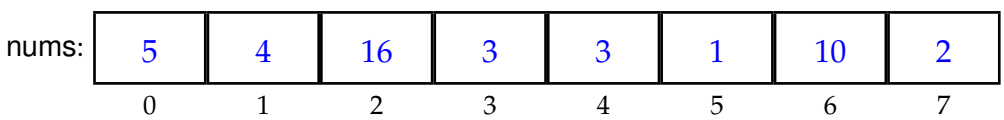
```
10
7
3
0:  5, 3:  3, 6:  2,
```

(b) [8 marks] Suppose that the variable nums is declared as before:

```
int [ ] nums = {5, 4, 9, 3, 1, 10, 2, 6};
```

Show the contents of nums after the following modify method is called on nums: modify(nums).

```
public void modify(int[ ] nums){
  nums[2] = 12 + nums[1];
  for( int i = nums.length −1; i > 3; i−−){
      nums[i] = nums[i−1];
  }
}
```



(Question 3 continued on next page)

(Question 3 continued)

(c) [10 marks] Complete the following average method. average has one parameter – an array of ints – and should return the average value of all numbers in the array. You may assume that the length of the array is greater than 0.

```
public int average(int [ ] nums){
    int ans = 0;
    int count = nums.length;

    for( int i = 0; i < nums.length; i++){
        ans = ans + nums[i];
    }
    return ans/count;




}
```

(d) [10 marks] Complete the following delete method. The delete method has two parameters – an array of ints and an index – and should delete the number at the index. Any numbers after the one deleted should be shifted one down, so the empty place is shifted to the end of the array and it should be filled with a special value -1.

```
public void delete(int [ ] nums, int index){
    for ( int i = index; i < nums.length−1; i++) {
      nums[i] = nums[i+1];
    }
    nums[nums.length−1]=−1;









}
```

(Question 3 continued on next page)

(Question 3 continued)

(e) [10 marks] What will the following fragment of Java print out?

```
int [][] data = {{1, 5, 21},
                 {3, 6, 14},
                 {3, 2, 21},
                 {3, 6, 14}};


System.out.println(data [0][2]);
System.out.println(data [1][1] + data [3][2]);


for( int i = 0; i < 4; i++){
   int s = 0;
   for( int j = 0; j < 3; j++){
      System.out.print(data[ i ][ j ]+ "  ");
      s = s + data[ i ][ j ];
   }
   System.out.println(s);
}
```

```
21
20
1  5  21  27
3  6  14  23
3  2  21  26
3  6  14  23
```

Student ID: .....

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 [17 marks]

(a) [8 marks] The following recursive method, recursiveDraw, draws circles on the canvas.

```
public void recursiveDraw(int x, int y, int s, int num){
    if (num > 0){
      this.canvas.drawOval(x, y, s, s);
      this.recursiveDraw(x + s, y, s * 2, num - 1);
    }
}
```

In the box below, sketch what it would draw if it were called by the statement:

recursiveDraw(100, 100, 10, 4);

Label the circles, e.g. Circle 1, Circle 2, etc., and specify the position and size of each circle, e.g. Circle 1: topLeft (100,100), size 10.

```
at (100, 100), size 10
at (110, 100), size 20
at (130, 100), size 40
at (170, 100), size 80
```

(Question 4 continued on next page)

(Question 4 continued)

(b) [9 marks] What will be printed out if the following recursive method is called by the statement:
printRec(5);

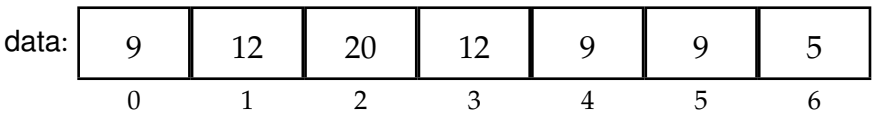
```
public void printRec(int num){
    if (num == 1)
        System.out.println(num);
    else {
        System.out.println("a: " + num);
        this.printRec(num - 1);
        System.out.println("b: " + num);
    }
}
```

```
a:  5
a:  4
a:  3
a:  2
1
b:  2
b:  3
b:  4
b:  5
```

Question 5. Debugging and Writing Loops [32 marks]

The following findDuplicates method is intended to find pairs of duplicate numbers in an array of numbers. For each pair of duplicate numbers, it should print out their indexes and the value of the duplicate number. It should also print out the total number of pairs at the end.

For example, for the array



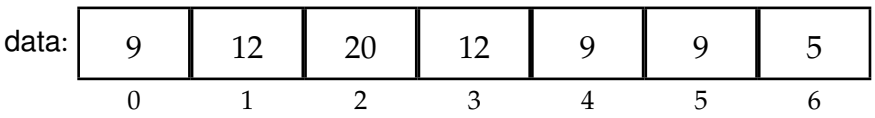
findDuplicates should print out:

```
same: 0&4:9
same: 0&5:9
same: 1&3:12
same: 4&5:9
number of pairs: 4
```

The version of findDuplicates below has several errors.

```
public void findDuplicates( int [] data) {
    int c = 1;
    for ( int i = 0; i < data.length; i++){
        for( int j = i ; j < data.length; j++){
            if (data[i] == data[j]){
                c++;
                System.out.println("same: "+ i + "&" +j+":"+ data[i]);
            }
        }
        System.out.println("number of pairs: " + c);
    }
}
```

(a) [10 marks] What does the findDuplicates method on the facing page print out if it is called with an argument that is the following array of numbers:




```
same:  0&0:9
same:  0&4:9
same:  0&5:9
number of pairs:  4
same:  1&1:12
same:  1&3:12
number of pairs:  6
same:  2&2:20
number of pairs:  7
same:  3&3:12
number of pairs:  8
same:  4&4:9
same:  4&5:9
number of pairs:  10
same:  5&5:9
number of pairs:  11
same:  6&6:5
number of pairs:  12
```

(b) [10 marks] Fix the errors in the version of findDuplicates on the facing page, so the method does what it is supposed to do..

Write your answers by modifying the code on the facing page

[part **(c)** is on next page.]

(Question 5 continued on next page)

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] Write a new method countOccurrences. This method has one parameter – an array of ints – and it should print every distinct number and the number of occurrence of each distinct number.

For example, for the array

data:	9	12	20	12	9	9	5
	0	1	2	3	4	5	6

countOccurrences should print out:

```
 9: 3 times
12: 2 times
20: 1 times
 5: 1 times
```

```
public void countOccurrences(int [ ] data){




  for ( int i=0; i < data.length;i++) {
    int c =1;
    if (data[i]!=-1){
      for( int j =i+1; j < data.length; j++) {
        if (data[i]==data[j]){
            c++;
            data[j]=-1;
        }
      }
      System.out.println(data[i]+ ":" + c);
    }
  }









}
```

****

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.

(You may detach this page)

Brief and partial documentation of some classes and methods

```
PrintStream class:                                  // Note, System.out is a PrintStream object
 public PrintStream (File f)                        // Constructor, for printing to a file
 public void close()                                // Close the file (if it is wrapping a File object)
 public void print (String s)                       // Prints s with no newline
 public void print (int i)                          // Prints i with no newline
 public void print (double d)                       // Prints d with no newline
 public void println ()                             // Prints a newline
 public void println (String s)                     // Prints s followed by newline
 public void println (int i)                        // Prints i followed by newline
 public void println (double d)                     // Prints d followed by newline
 public void printf (String format, ...)            // Prints the format string, inserting the remaining
                                                    // arguments at the %'s in the format string:
                                                    //   %s for Strings.
                                                    //   %d for ints, (%3d: use at least 3 characters),
                                                    //   %.2f for 2dp doubles,
                                                    //   (%6.2f: at least 6 characters and 2dp),
                                                    // Use \n for newline
Scanner class:
   public Scanner (InputStream i)                   // Constructor. Note: System.in is an InputStream
   public Scanner (File f)                          // Constructor, for reading from a file
   public Scanner (String s)                        // Constructor, for reading from a string
   public boolean hasNext()                         // Returns true if there is more to read
   public boolean hasNextInt()                      // Returns true if the next token is an integer
   public boolean hasNextDouble()                   // Returns true if the next token is a number
   public String next()                             // Returns the next token (chars up to a space/line)
                                                    //   (throws exception if no more tokens)
   public String nextLine()                         // Returns string of chars up to next newline
                                                    //   (throws exception if no more tokens)
   public int nextInt()                             // Returns the integer value of the next token
                                                    //   (throws exception if next token is not an integer
                                                    //     or no more tokens)
   public double nextDouble()                       // Returns the double value of the next token
                                                    //   (throws exception if next token is not a number
                                                    //     or no more tokens)
   public void close()                              // Closes the file (if it is wrapping a File object)

File class:
   public File (String fname)                       // Constructor. Creates a File object attached to the
                                                    //   file with the name fname
   public boolean exists()                          // Returns true iff the file exists

Integer class:
   public static final int MAX_VALUE                // The largest possible int: 2^(31−1)
   public static final int MIN_VALUE                // The smallest possible int: −2^(31)
```

(Continued on next page)

String **class**:

```
public int length()                          // Returns the length (number of characters) of the string
public boolean equals(String s)              // String has same characters as s
public boolean equalsIgnoreCase(String s)    // String has same characters as s, ignoring their case
public String toUpperCase(String s)          // Returns upper case copy of string
public String toLowerCase(String s)          // Returns lower case copy of string
public boolean startsWith(String s)          // First part of string matches s
public boolean contains(String s)            // s matches some part of the string
public String substring(int j, int k)        // Returns substring from index j to index k−1
public int indexOf(String s)                 // Returns the index of where s first matches
                                             // Returns −1 if string does not contain s anywhere
```

Math **class**:

```
public static double sqrt(double x)          // Returns the square root of x
public static double min(double x, double y) // Returns the smaller of x and y
public static double max(double x, double y) // Returns the larger of x and y
public static double abs(double x)           // Returns the absolute value of x
public static int min(int x, int y)          // Returns the smaller of x and y
public static int max(int x, int y)          // Returns the larger of x and y
public static int abs(int x)                 // Returns the absolute value of x
```

DrawingCanvas **class**:

```
public void clear()                                  // Clears the drawing canvas
public void setForeground(Color c)                   // Change the colour for later commands
public void drawLine(int x, int y, int u, int v)     // Draws line from (x, y) to (u, v)
public void drawRect(int x, int y, int wd, int ht)   // Draws outline of rectangle
public void fillRect(int x, int y, int wd, int ht)   // Draws solid rectangle
public void clearRect(int x, int y, int wd, int ht)  // Draws clear rectangle
public void drawOval(int x, int y, int wd, int ht)   // Draws outline of oval
public void fillOval(int x, int y, int wd, int ht)   // Draws solid oval
```

Color **class**:

```
public Color(int red, int green, int blue)   // Make a colour; arguments must be 0..255
Color.gray, Color.blue, Color.red,           // Some of the predefined colours
Color.green, Color.black, Color.white
```

MouseListener **interface**:

```
public void mousePressed(MouseEvent e);      // Called when mouse pressed
public void mouseReleased(MouseEvent e);     // Called when mouse released
public void mouseClicked(MouseEvent e);      // Called when mouse clicked
public void mouseEntered(MouseEvent e);      // Called when mouse enters component
public void mouseExited(MouseEvent e);       // Called when mouse exits component
```

MouseEvent **class**:

```
public int getX()                            // Get the x component of the mouse position
public int getY()                            // Get the y component of the mouse position
```

(You may detach this page)

Small fragments of programs with examples of Java syntax

```
public void prompt(){
    Scanner sc = new Scanner(System.in);
    System.out.print("answer: ");
    if (sc.hasNextInt()){
        int n = sc.nextInt();
        System.out.println(n);
    }
    else{
        String s = sc.next();
        System.out.println(s);
    }
}
```

```
public void table( int n){
    for ( int i=1; i<=n; i++){
        for ( int j=1; j<=n; j++){
            System.out.printf(" %3d |", i*j);
        }
        System.out.println();
    }
}
```

```
public String longest(String fname){
    String[] words = new String[20];
    try{
        Scanner f = new Scanner(new File(fname));

        int i =0;
        while (f.hasNext() && i < words.length){
            words[i] = f.next();
            i++;
        }
    }
    catch(Exception e){System.out.println("File broken: "+ e);}

    String ans = "";
    for ( int j=0; j<words.length; j++ ){
        if ( words[j].length() > ans.length() ){
            ans = words[j];
        }
    }
    return ans;
}
```

(Continued on next page)

```
public void actionPerformed(ActionEvent e){
    String button = e.getActionCommand();
    if ( button.equals("Clear") )
        this.canvas.clear();
    else if ( button.equals("Quit"))
        frame.dispose();
}
```

```
public void mouseClicked(MouseEvent e){
    this.canvas.fillOval(e.getX(), e.getY(), 10, 10);
}
```

```
public class Flag{
    private int size = 40;
    private int x;
    private int y;

    public Flag(int u, int v){
        this.x = u;
        this.y = v;
    }

    public void draw(DrawingCanvas canvas){
        canvas.setColor(Color.black);
        int left = this.x - this.size/2;
        int top = this.y - this.size/2;
        canvas.drawRect(left, top, this.size, this.size);
        canvas.setColor(Color.green);
        canvas.drawLine(this.x, top, this.x, top+this.size);
        canvas.drawLine(left, this.y, left+this.size, this.y);
    }
}
```

```
  public void print2Darray(int [][] table){
    for (int row=0; row<table.length; row++){
      for (int col=0; col<table[row].length; col++){
        System.out.printf("%4d", table[row][col]);
      }
      System.out.println();
    }
  }
```
