

Family Name:

Other Names:

ID Number:

COMP102: Test 1

14 March, 2008

Instructions

- Time allowed: **45 minutes** .
- Answer **all** the questions. There are 45 marks in total.
- Write your answers in the boxes in this test paper and hand in all sheets.
- If you think some question is unclear, ask for clarification.
- The last page contains some Java documentation and model solutions for part of Assig 2.
- This test will contribute 4% of your final grade, but only if it helps your grade.
- Paper translation dictionaries and calculators without a full set of alphabet keys are permitted.

Questions

Marks

1. Basic Java	[9]	<input type="text"/>
2. Understanding variables	[5]	<input type="text"/>
3. Defining a Method	[10]	<input type="text"/>
4. Using a Scanner and println	[14]	<input type="text"/>
5. Loops and Conditionals	[7]	<input type="text"/>
	TOTAL:	<input type="text"/>

Please answer the following question. (Your answer will not affect your mark in any way.)

How much programming had you done before starting the course?

Little or none

Some (used variables, if's, and loops)

Lots (used arrays, defined methods/functions with parameters, used libraries)

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

[9 marks]

For each of the following ten terms, find a corresponding element of the program below, and draw a labelled circle around the element. The first one is done as an example.

1. Class name
2. A string
3. A variable
4. An expression with an operator
5. A comment
6. The name of a method
7. The name of a type
8. An assignment statement
9. A method call
10. A parameter declaration

```

public class Question1{
    /** Computes the amount of paint for a wall */

    public void computeSize(){
        System.out.print("area: ");

        Scanner sc = new Scanner(System.in);

        double area = sc.nextDouble();

        double paint = area / 5.4 ;    // 5.4 square meters per litre

        System.out.println("Wall needs " + paint + " litres");
    }

    public void checkName(String name){
        if ( name.equals("George Bush") ) {
            System.out.print("Entry Denied");
        }
    }
}

```

Question 2. Understanding variables

[5 marks]

Suppose the following `assignToVars` method is called with an argument of 8, (eg, you call the method using BlueJ and enter 8 in the dialog box asking for the value of `n`). What will it print out?

```
public void assignToVars(int n){
    System.out.println("assignToVars (" + n + ") :");

    int a = n + 4;
    int b = n + a;
    System.out.println("a is: " + a);
    System.out.println("b is: " + b);

    a = a - 5;
    b = a * 2;
    b = b - 1;

    System.out.println("a is now: " + a);
    System.out.println("b is now: " + b);

    System.out.printf("was %d, now %d, will be \n", a-b, n);
}
```

```
assignToVars (8) :
```

Question 3. Defining a Method

[10 marks]

(a) [7 marks] Suppose we need to organise the students in a course into teams for a programming competition. Two students must be the manager and deputy manager for the competition (and therefore can't be in a team), and each team has to have exactly 5 members. Complete the following `computeTeams` method so that it prints out the number of programming teams that can be made for a given number of students. The method has one parameter — an integer specifying the number of students in the course.

The method should print out the result in a form such as:

```
"A course with 50 students can have 9 teams".
```

or

```
"A course with 12 students can have 2 teams".
```

(where the actual numbers depend on the argument value passed to the method.)

```
public void computeTeams(int numStudents){
```

```
}
```

(b) [3 marks] Modify your program above so that it also handles the additional requirement that if the class has 100 students or more, there has to be an assistant manager also, who cannot be a member of a team.

[Put your answer to (b) in the box above, or use the spare pages (p2 or p10)]

Question 4. Using a Scanner and println

[14 marks]

(a) [6 marks] Consider the following method which will prompt the user for some values and print something out.

Hint: draw boxes for each variable and write its value.

```
public void describeCourse(){
    Scanner scan = new Scanner(System.in);
    System.out.print("Enter discipline, code, points: ");
    String d = scan.next();
    int c = scan.nextInt();
    int p = scan.nextInt();

    System.out.print("Name of course: ");
    String m = scan.next();
    String n = scan.nextLine();

    System.out.println("Course = " + d + c);
    System.out.println("Name = " + n);

    String description = n + m;

    System.out.println("Line1 = " + description);
    System.out.println("Line2 = " + c + p);
    System.out.println("Line3 = " + (c + p));
}
```

What will the method print out if the user typed the following answers in response to the prompts:

Enter discipline, code, points: COMP 303 15

Name of course: Analysis of Algorithms

(Question 4 continued on next page)

(Question 4 continued)

(b) [8 marks] Complete the following method to print out a library overdue notice. The method should first ask the user to enter the title of the book, then the number of days overdue. It should then print out a notice like the one shown below. It should use a Scanner to read the title, name, and days overdue from the user.

For example, if the user typed `Gulliver's Travels` for the title, and `18` for the number of days, the output should look something like:

```
Your copy of Gulliver's Travels  
is now 18 days overdue.  
Please return it now.
```

```
public void overdueNotice(){
```

```
}
```

Question 5. Loops and Conditionals (harder)

[7 marks]

(a) [5 marks] What will the following method print out if it is called with the arguments 14 and 6?

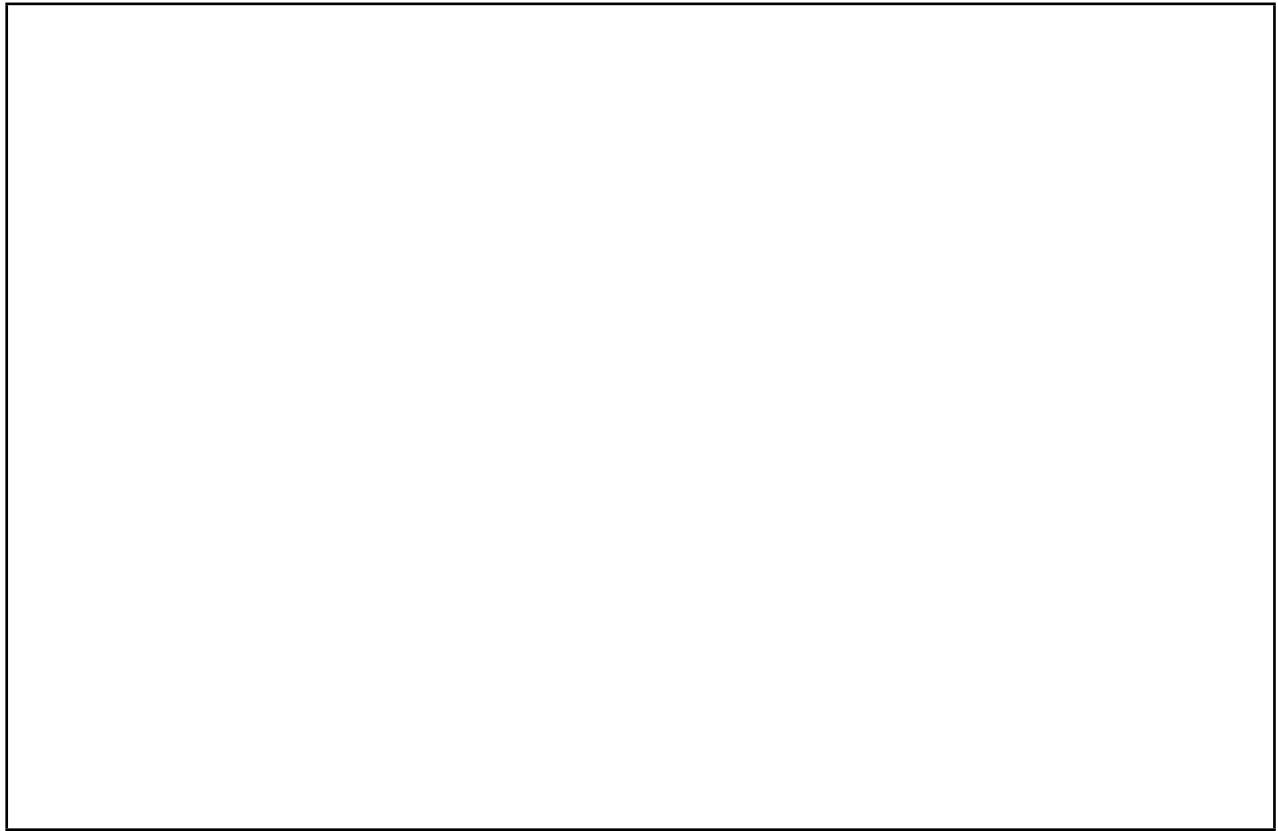
```
public void compute(int a, int b){
    System.out.printf("compute(%d, %d)\n", a, b);
    int x = a;
    int y = b;
    int ans = 0;
    while ( (x > 0) && (y > 0) ){
        System.out.println(x + ", " + y);
        ans = ans + 1;
        if (x >= y){
            x = x-y;
            y = y-1;
        }
        else {
            int temp = x;
            x = y;
            y = temp;
        }
    }
    System.out.println("ans = "+ ans);
}
```

compute(14, 6)

(Question 5 continued on next page)

(Question 5 continued)

(b) [2 marks] Explain why the compute method is always guaranteed to stop, whatever its input.



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.

Brief and partial documentation of some classes and methods

PrintStream class: *// Methods you can call on System.out*

```
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
                                                 // %6.2f for doubles, with 2dp and at least 6 characters ),
                                                 // \n for newline
```

Scanner class: *// Methods you can call on a Scanner object*

```
public Scanner(InputStream i);                // Constructor. eg new Scanner(System.in)
public String next ();                        // Returns the next token ( characters up to a space/line )
public String nextLine ();                    // Returns the rest of the current line
public int nextInt ();                        // Returns the integer value of the next token
                                                 // (error if next token is not an integer)
public double nextDouble ();                // Returns the double value of the next token
                                                 // (error if next token is not a number)
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 void close ();                        // Closes the file ( if it is wrapping a File object )
```

DrawingCanvas class: *// Methods you can call on a DrawingCanvas object*

```
public void clear ();                        // Clears the drawing canvas
public void setColor(Color c);               // Change the colour for later commands
public void drawLine(int x, int y, int u, int v);    // Draws line from cd{(x, y) to cd{(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
```