Family Name: . . . . . **Other Names:** . . . . . .

ID Number: . . . . . .

COMP102: Test 1

27 July, 2007

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.
- There is some Java documentation at the end of the test paper.
- Model solutions for part of Assignment 2 are also included at the end of the test paper.
- This test will contribute 5% of your final grade, but only if it helps your grade.
- Non-electronic translation dictionaries and calculators without a full set of alphabet keys are permitted.

Questions	**Marks**	
1. Basic Java	[8]	
2. Understanding variables	[5]	
3. Defining a Method	[10]	
4. Using a Scanner and println	[15]	
5. Loops and Conditionals	[7]	
	TOTAL:	

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.

Student ID: .....

Question 1. Basic Java [8 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. An expression
4. A comment
5. A method header
6. A name of a type
7. A variable declaration
8. An assignment statement
9. A method call

```
public class Test {                                          1

    /** Computes the volume of a box of variable width */
    public void computeSize(){
        System.out.print("width: ");
        Scanner scan = new Scanner(System.in);
        int width = scan.nextInt ();
        int size = width * 20 + 40;
        System.out.println("Size is " + size);
    }
    public void doubleUp(String name){
        String item = "cheese";
        System.out.printf("A %s %s  %s please", name, name, item);
    }
}
```

Question 2. Understanding variables [5 marks]

Suppose the following jumble method is called with an argument of 12, (eg, you call the method using BlueJ and enter 12 in the dialog box asking for the value of n). What will it print out?

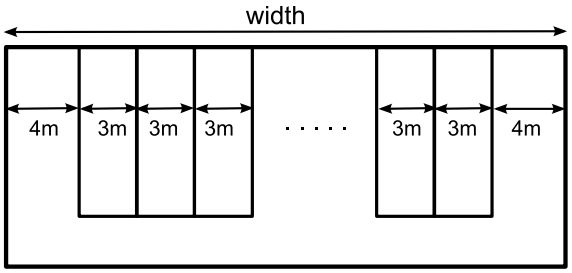
```
public void jumble(int n){
    System.out.println("jumble(" + n + "):");

    int a = n * 2;
    System.out.println("a is: "+ a);
    int b = n - 2;
    System.out.println("a is now: "+ b);

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

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

    System.out.printf("a %d b %d is ", n, a+b);
}
```

```
jumble(12):
```

Question 3. Defining a Method [10 marks]

(a) [7 marks] Complete the following computeLanes method so that it prints out the number of bowling lanes that will fit in a building of a specified width. The method should have one parameter — an integer specifying the width of the building in meters.

Each lane is 3 meters wide. There has to be an exit corridor of at least 4 meters wide against each wall of the building:



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

```
"A building 33 meters wide holds 8 lanes".
```

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

```
public void computeLanes(int width){




}
```

(b) [3 marks] Modify your program above so that it also handles the additional requirement that whenever the building has more than 10 lanes, there has to be an additional exit corridor 2 meters wide in the middle of the building.

(Put your answer to (b) in the box above)

Question 4. Using a Scanner and println [15 marks]

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

```
public void bankPayment(){
    Scanner scan = new Scanner(System.in);
    System.out.print("Enter bank, payee name, code, and amounts: ");
    String b = scan.next();
    String p = scan.nextLine();

    System.out.println("bank = " + b);
    System.out.println("payee = " + p);

    String message = p + b;
    System.out.println("line1 = " + message);


    int c = scan.nextInt();
    int a1 = scan.nextInt();
    int a2 = scan.nextInt();
    String rest = scan.nextLine();

    System.out.println("line2 = " + c + a1);
    System.out.println("line3 = " + (c + a2));
    System.out.println("line4 = " + c + rest);
}
```

What will the method print out if the user typed the following lines in response to the prompt:

```
National Bank Jane Smith
4519 21 35 14 18
```

(Question 4 continued on next page)

(Question 4 continued)

(b) [9 marks] Complete the following doorPlate method so that it first asks the user to enter a room number, then asks for their name, and then prints out a sign for their office door like one shown below. It should use a Scanner to read the number and name from the user.

If the user typed the number 423 and the name "Chris Hughes", the output should look something like:

```
Dr Chris Hughes, Computer Science
         Cotton 423
     School of Engineering
```

```
public void doorPlate(){









}
```

Question 5. Loops and Conditionals (harder) [7 marks]

(a) [5 marks] What will the following method print out if it is called with the argument 24?

```
public void compute(int n){
    System.out.printf("compute(%d)\n", n);
    int x = 0;
    int y = n;
    while (y > x){
        System.out.printf("%d : %d\n", x, y);
        if (y % 2 == 0) {
            x = x * 2;
            y = y - 1;
        }
        else {
            x = x + 2;
            y = y - 3;
        }
    }
    System.out.printf("ans is %d\n", x);
}
```

```
compute(24)
```

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

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.

****

You may tear off this sheet

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
                                      //   %10s  for Strings, using at least 10 characters
                                      //   %d    for ints
                                      //   %3d   for ints, using at least 3 characters
                                      //   %.2f  for doubles, with 2dp
                                      //   %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 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 (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 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
```