COMP102: Test 1

8 Aug, 2013

Model Solutions

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.

- Brief Java documentation is provided with the test

- This test contributes 15% of your final grade
  (But your mark will be boosted up to your exam mark if that is higher.)

- You may use paper translation dictionaries, and calculators without a full set of alphabet keys.

- You may write notes and working on this paper, but make sure your answers are clear.

Questions	Marks	
1. Components of Java Programs	[8]	
2. Understanding programs with variables	[8]	
3. Defining a Method	[8]	
4. Methods, Arguments, and Parameters	[6]	
5. Using Objects	[8]	
6. Defining and Calling Methods	[7]	
	TOTAL:	

Question 1. Components of Java Programs   [8 marks]

The questions on the facing page refer to the code below.

```java
1   import comp102.*;
2   public class Question1{
3
4       public static final String message= "welcome to COMP102!";
5
6       public void showMessage(){
7           String name = UI.askString("Enter your name: ");
8           String msg = this.getMessage(name);
9           this.drawMessage(100, 100, msg);
10      }
11
12      public String getMessage(String n){
13          return n + ",  " + message;
14      }
15
16      public void drawMessage(double x, double y, String lab){
17          double right  = x+200;
18          UI.drawLine(x, y,  right , y);
19          UI.drawLine(x, y+2, right , y+2);
20          UI.drawString(lab, x+10, y−2);
21      }
22
23  }
```

(Question 1 continued)

The following questions refer to the code on the facing page:

(a) [1 mark] Write the name of the class.

> Question1

(b) [1 mark] List the names of the methods.

> showMessage, getMessage, drawMessage

(c) [1 mark] What <u>type</u> of value is returned by the getMessage method?

> String

(d) [1 mark] List the <u>variables</u> that have the String type.

> name, msg

(e) [1 mark] How many <u>arguments</u> does the drawMessage method require?

> 3

(f) [1 mark] List the names of all the <u>parameters</u> declared in the program?

> n, x, y, lab

(g) [1 mark] List the lines that contain a <u>call to a method</u>?.

> 7, 8, 9,18, 19,20

(h) [1 mark] List the arguments in line 9.

> 100, 100, msg

Question 2. Understanding programs with variables [8 marks]

Consider the process method below. In the box on the facing page (p5), write the output that process would produce. Note that the first part of each line of output is provided for you.

Hint: Keep track of the value of each variable in the boxes.

```
public void process(){
    int  level = 1;

    int  first  = 10;
    int  second = 20;
    int  third  = 30;
    UI. println ("level is " + level);
    UI. println ("one  is " + first );
    UI. println ("two  is " + second);
    UI. println ("three is " + third);
    UI. println ("------");

    level  = level  + 1;
    first    = second;
    second = third;

    UI. println ("level is " + level);
    UI. println ("one  is " + first );
    UI. println ("two  is " + second);
    UI. println ("three is " + third);
    UI. println ("------");

    level  = level  + 1;
    first  = first  + second;
    second = third  + second;
    third  = third  + second;

    UI. println ("level is " + level);
    UI. println ("one  is " + first );
    UI. println ("two  is " + second);
    UI. println ("three is " + third);
}
```

level:

first:

second:

third:

(Question 2 continued)

```
level is 1

one    is 10

two    is 20

three is 30

------

level is 2

one    is 20

two    is 30

three is 30

------

level is 3

one    is 50

two    is 60

three is 90
```

Question 3. Defining a Method

Complete the following drawTarget() method for drawing a target logo shown below.

The method should
- ask the user for the x position of the center of the logo.
- ask the user for the y position of the center of the logo.
- ask the user for the size of the logo, which is the diameter of the biggest circle.
- work out the coordinates and dimensions. The ratio of the diameters of the three circles is 3:2:1.
- draw the logo using the right color: red, white and then red.

```
public void drawTarget(){

    double x = UI.askDouble("Center x:");
    double y = UI.askDouble("Center y: ");
    double size = UI.askDouble("size: ");
    UI.setColor(Color.RED);
    UI. fillOval (x−size/2, y−size/2, size,size );
    UI.setColor(Color.WHITE);
    UI. fillOval (x−size/3, y−size/3, 2∗size/3, 2∗size/3);
    UI.setColor(Color.RED);
    UI. fillOval (x−size/6, y−size/6, size/3, size /3);




}
```

Question 4. Methods, Arguments, and Parameters [6 marks]

What will the following testNumbers method print in the text pane?
Note that testNumbers calls printNumbers which also prints some numbers.

```java
public void TestNumbers(){
    int z = this.printNumbers(6, 1);
    UI. println (z);
    UI. println ("--------");
    int x = 2;
    int y = 10;
    z = this.printNumbers(x, y);
    UI. println (z);
    UI. println ("---------");
    z = this.printNumbers(y, x);
    UI. println (z);
}

public int printNumbers(int s,  int t){
    s = s * 2;
    UI. println (s + ",  " + t);
    return s + t;
}
```

```
12, 1
13


-------------
4, 10
14
-------------
20, 2
22




```

Question 5. Using objects [8 marks]

Suppose the Tutor class has one constructor and two methods:

Constructor:

public Tutor(*String* name, *int* id, *String* note)
// constructs a Tutor object, with the specified name, id and a note.
//The note can be anything, but typically it is used to specify the
//student category, e.g. PhD, Master or undergraduate.

Methods:

public void addHours(*int* hour, *String* job)
// adds the working hours to the student's total working hours,
//and records the kind of job he did, e.g. lab, marking or tutorial.

public void printPayslip ()
// works out the total amount that needs to be paid and prints out the
// details of the tutor, the total hours, the jobs and the total payment.

Complete the following tutorPayment method, which should first create two Tutor objects, the first object for Alex Smith, whose ID is 12345 and he is a PhD student, and the second object for Bob Allen, whose ID is 67890 and he is an undergraduate student.

The method should then add the following hours for Alex and Bob.

1st object(Alex): 3 hours for marking
2nd object (Bob): 5 hours for labs
1st object(Alex): 10 hours for labs
1st object(Alex): 2 hours for tutorial

Finally, the method should print the payslip for both of them, first for Alex and then for Bob.

Please note: you must use the Tutor class and call the two methods in the Tutor class to do this.

(Question 5 continued)

```java
public void tutorPayment(){
    Tutor t1 = new Tutor("Alex Smith", 12345, "Phd");
    Tutor t2 = new Tutor("Bob Allen", 67890, "undergraduate");
    t1.addHours(3, "marking");
    t2.addHours(5, "labs");
    t1.addHours(10, "labs");
    t1.addHours(2, "tutorial");
    t1. printPayslip ();
    t2. printPayslip ();



}
```
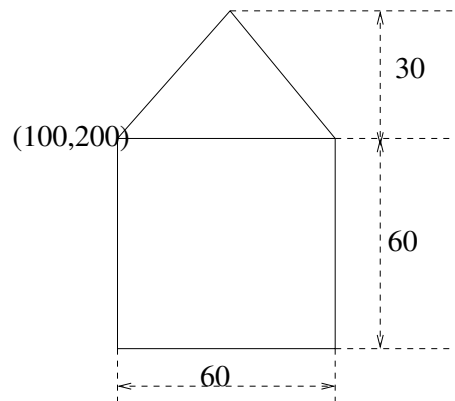
The following drawHouse1 method draws a house using a square and two lines.

```
public void drawHouse1(){
    UI.drawRect(100, 200, 60, 60);
    UI.drawLine(100, 200, 130, 170);
    UI.drawLine(160, 200, 130, 170);
}
```


The drawHouse1 method can only draw a house at a certain place in a fixed size because it uses many literal values. It would be better to define this method with parameters, so it can be called by other methods to draw houses at different places and in different sizes.

Define a new method drawHouse2 by modifying this method, so it uses parameters to specify the location and size. It should still draw a house using a square and two lines, and the height of the roof (triangle) should be half of the width of the house (the square).

```
public void drawHouse2(                                            ){

                              double x,  double y,  double size
    UI.drawRect(x, y,  size,  size);
    UI.drawLine(x, y,  x+size/2, y−size/2);
    UI.drawLine(x+size, y,  x+size/2, y−size/2);




}
```

Write a test method that called the drawHouse2 method three times to draw three different houses. You will need to work out the arguments yourself, and any reasonable location and size are acceptable.

```java
public void testDrawHouses(){


    this.drawHouse2(100, 200, 40);
    this.drawHouse2(300, 200, 100);
    this.drawHouse2(400, 200, 80);









}
```

****

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.