

Family Name:

Other Names:

ID Number:

Signature

COMP102: Test 1

8 Apr, 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]	<input type="text"/>
2. Understanding programs with variables	[8]	<input type="text"/>
3. Defining a Method	[10]	<input type="text"/>
4. Methods, Arguments, and Parameters	[6]	<input type="text"/>
5. Using Objects	[6]	<input type="text"/>
6. Defining and Calling Methods	[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 (eg, used arrays, defined methods/functions with parameters, used libraries)

Question 1. Components of Java Programs

[8 marks]

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

```
1 import comp102.*;
2 public class Question1{
3
4     public void insuranceClaim(){
5         int vehicleCategory = UI.askInt("Vehicle Type: ");
6         double repairCost = this.computeCost(vehicleCategory);
7         double cover = UI.askDouble("Maximum coverage available: ");
8         this.reportClaim(repairCost, cover);
9     }
10
11    public double computeCost(int vType){
12        double parts = UI.askDouble("Cost of parts: ");
13        double partsCost = 200 + parts * 1.5;
14        double labour = 500 * vType;
15        return (partsCost + labour);
16    }
17
18    public void reportClaim(double cost, double cover){
19        UI.printf ("Estimated cost of repair is $%4.2f\n", cost);
20        UI.printf ("Payment will be $%4.2f\n", Math.min(cover, cost));
21    }
22
23 }
```

(Question 1 continued)

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

(a) [1 mark] Which line or lines contain the header of a class? (list the line numbers.)

2

(b) [1 mark] Which line or lines contain the header of a method definition?

4, 11, 18

(c) [1 mark] What type of value is returned by the computeCost method?

double

(d) [1 mark] Which line or lines contain a literal String?

5, 7, 12, 19, 20

(e) [1 mark] How many arguments does the reportClaim method require?

two

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

vType, cost, cover

(g) [1 mark] List four lines that contain a call to a method?

5, 6, 7, 8, 12, 19, 20

(h) [1 mark] Which line or lines contain an expression containing an operator?

13, 14, 15

Question 2. Understanding programs with variables

[8 marks]

Consider the processLevels method below. In the box on the facing page (p5), write the output that processLevels 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 processLevels(){
```

```
    int level = 5;
```

```
    String first = "a";
```

```
    String second = "b";
```

```
    String third = "c";
```

```
    UI.println ("level is " + level);
```

```
    UI.println ("one is " + first );
```

```
    UI.println ("two is " + second);
```

```
    UI.println ("three is " + third);
```

```
    UI.println ("-----");
```

```
        first  = second;
```

```
    second = third;
```

```
    third = first ;
```

```
    level = level - 1;
```

```
    UI.println ("level is " + level);
```

```
    UI.println ("one is " + first );
```

```
    UI.println ("two is " + second);
```

```
    UI.println ("three is " + third);
```

```
    UI.println ("-----");
```

```
        first = first + ":X";
```

```
    second = third + ":" + second;
```

```
    third = second + third;
```

```
    level = level / 8;
```

```
    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 5

one is a

two is b

three is c

level is 4

one is b

two is c

three is b

level is 0

one is b:X

two is b:c

three is b:cb

Question 3. Defining a Method

[10 marks]

Fill-em-up Dentists are a discount dental practice with a simple set of charges. They charge \$30 for the first visit, \$20 for every follow-up visit, plus \$40 per filling.

For example, for a treatment plan with 3 follow-up visits and 4 fillings, the total cost would be \$250 [= \$30 + (3 × \$20) + (4 × \$40)].

(a) [6 marks] Complete the following fillEmUp() method for calculating the estimated cost of treatment. The method should

- ask the user for the number of follow-up visits
- ask for the number of fillings
- calculate and print the cost of the fillings and the total estimated cost, (including the initial visit).

For example, if the user entered 3 visits and 4 fillings, the interaction should be the following (where the user's input is in bold):

```
Follow-up visits: 3
Fillings: 4
Cost of fillings = $160
Total cost = $250
```

```
public void fillEmUp(){
    int visits = UI.askInt("Follow-up visits: ");
    int fillings = UI.askInt("Fillings: ");
    int fillingCost = fillings *40;
    UI.println ("Cost of fillings = $" + fillingCost);
    int totalCost = 30 + ( visits * 20) + fillingCost ;
    UI.println ("Total cost = $" + totalCost);

}
```

(b) [4 marks] Suppose the dentists decide to modify their charges, by adding an additional charge of \$20 when there is only one filling, and also offer a discount of 10% on the total cost if there are 4 or more follow-up visits.

Complete the following `newFillEmUp` method for computing the updated charges.

The method should just print the total cost (it does not need to print the cost of the fillings)

```
public void newFillEmUp(){
    int visits = UI.askInt("Follow-up visits: ");
    int fillings = UI.askInt("Fillings: ");
    int fillingCost = fillings *40;
    int totalCost = 30 + ( visits * 20) + fillingCost ;
    if ( fillings == 1) {
        totalCost = totalCost + 20;
    }
    if ( vists > 3){
        totalCost = totalCost * 0.9;
    }
    UI.println ("Total cost = $" + totalCost);
}
```


Question 4. Methods, Arguments, and Parameters

[6 marks]

What will the following listStaff method print in the text pane?
Note that listStaff calls printPerson which is also defined below.

```
public void listStaff (){
    String expertise = "Databases";
    String phone = "1234";
    UI.println ("Current library staff:");
    this.printPerson("Jack", "4321", "Reference");
    this.printPerson("Jim", phone, expertise);
    String name = "Jane";
    expertise = "Restocking";
    this.printPerson(expertise, "3535", name);
    this.printPerson("Julia", expertise, phone);
}
```

expertise:

phone:

name:

```
public void printPerson(String name, String phone, String exp){
    UI.println (name + " (" + exp + ")");
    UI.println (" phone: " + phone);
    UI.println ("---");
}
```

name:

phone:

exp:

```
Current library staff:
Jack (Reference)
phone: 4321
---
Jim (Databases)
phone: 1234
---
Restocking (Jane)
phone: 3535
---
Julia (1234)
phone: Restocking
---
```

Question 5. Using objects

[6 marks]

Suppose the Bus class has one constructor and two methods:

Constructor:

```
public Bus(int capacity)
    // constructs a Bus object , with the specified capacity , and displays it
    // at the left edge of the graphics pane, facing to the right .
```

Methods:

```
public void move(double dist)
    // makes the bus move forward by the specified distance in the current direction .
```

```
public void turn(double angle)
    // makes the Bus turn by the given angle (eg 90 to turn right , -90 to turn left )
```

Complete the following drive method, which should first create two Bus objects, one with capacity for 40 passengers, and one with capacity for 33 passengers. It should then make the first bus move forward 30 units and turn right, and then make the second bus move forward 20 units and turn left. It should then make each bus move forward a further 50 units.

```
public void drive(){
    Bus bus1 = new Bus(40);
    Bus bus2 = new Bus(33);
    bus1.move(30);
    bus1.turn(90);
    bus2.move(20);
    bus2.turn(-90);
    bus1.move(50);
    bus2.move(50);
```

```
}
```

Question 6. Defining and Calling Methods

[7 marks]

The following singAJingle method prints out a the words of a repetitive song.

```
public void singAJingle(){
    Ul.println ("Peter Piper picked 10 green peppers, ");
    Ul.println ("5 red peppers, Mr Piper picked, ");
    Ul.println ("Peter picked 20 orange peppers for Mrs Piper.");
    Ul.println ();
    Ul.println ("James Johnson picked 7 green peppers, ");
    Ul.println ("2 red peppers, Mr Johnson picked, ");
    Ul.println ("James picked 17 orange peppers for Mrs Johnson.");
    Ul.println ();
    Ul.println ("Fred Fulsome picked 12 green carrots, ");
    Ul.println ("7 red carrots, Mr Fulsome picked, ");
    Ul.println ("Fred picked 22 orange carrots for Mrs Fulsome.");
    Ul.println ();
    Ul.println ("Kris Kringle picked 9 green potatoes, ");
    Ul.println ("4 red potatoes, Mr Kringle picked, ");
    Ul.println ("Kris picked 19 orange potatoes for Mrs Kringle.");
    Ul.println ();
    Ul.println ("Craig Carson picked 3 green beans, ");
    Ul.println ("0 red beans, Mr Carson picked, ");
    Ul.println ("Craig picked 13 orange beans for Mrs Carson.");
    Ul.println ();
}
```

(Question 6 continued)

The `singAJingle` method is not well designed: it has quite a bit of repetition and many literal values. It would be better design to define another method called `printVerse` which prints out a single verse (computing some of the numbers) and make `singAJingle` call the `printVerse` method four times, as in the version of `singAJingle` below.

Complete the definitions of `singAJingle` and `printVerse` below so that they print out the same song. You will need to determine the appropriate arguments for the calls to `printVerse`, the appropriate parameters in the header of `printVerse`, and the statements in the body of `printVerse`. You should not use more than four parameters.

Note: Nobody can pick a negative number of any vegetable!

```
public void singAJingle(){

    this.printVerse( "Peter", "Piper", 10, "peppers");

    this.printVerse( "James", "Johnson", 7, "peppers");

    this.printVerse( "Fred", "Fulsome", 12, "carrots");

    this.printVerse( "Kris", "Kringle", 9, "potatoes");

    this.printVerse( "Craig", "Carson", 3, "beans");
}

public void printVerse( String nm1, String nm2, int count, String veg){

    if (count < 0){
        count = 0;
    }
    UI.printf ("%s %s picked %d green %s\n", nm1, nm2, count, veg);
    int countRed = count - 5;
    if (countRed < 0){
        countRed = 0;
    }
    UI.printf ("%d red %s, Mr %s picked\n", countRed, veg, nm2);
    UI.printf ("%s picked %d orange %s for Mrs %s\n", nm1, count+10, veg, nm2 );
    UI.println ();

}
```

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.