

Family Name:..... Other Names:

ID Number: Signature

Model Solutions

COMP 102: Test 1

6 April, 2016

Instructions

- Time allowed: **50 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 alpha-
bet keys.
- You may write notes and working on this paper, but make sure your answers are clear.

Questions

Marks

1. Understanding programs

[12]

2. Writing programs with input, output and **if**

[9]

3. Writing methods that use objects

[7]

4. Understanding arguments and parameters

[7]

5. Writing methods with **if** and **while**

[10]

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.

Question 1. Understanding programs

[12 marks]

(a) [4 marks] Understanding variables

What will the following method print out?

```
public void printStuff (){  
    int x = 5;  
    int y = 10;  
    int z = x + y;  
    Ul.println ("Line 1: " + (x+1));  
    Ul.println ("Line 2: " + z);  
    x = x + 3;  
    y = y + x;  
    Ul.println ("Line 3: "+ y);  
    Ul.println ("Line 4: " + z);  
}
```

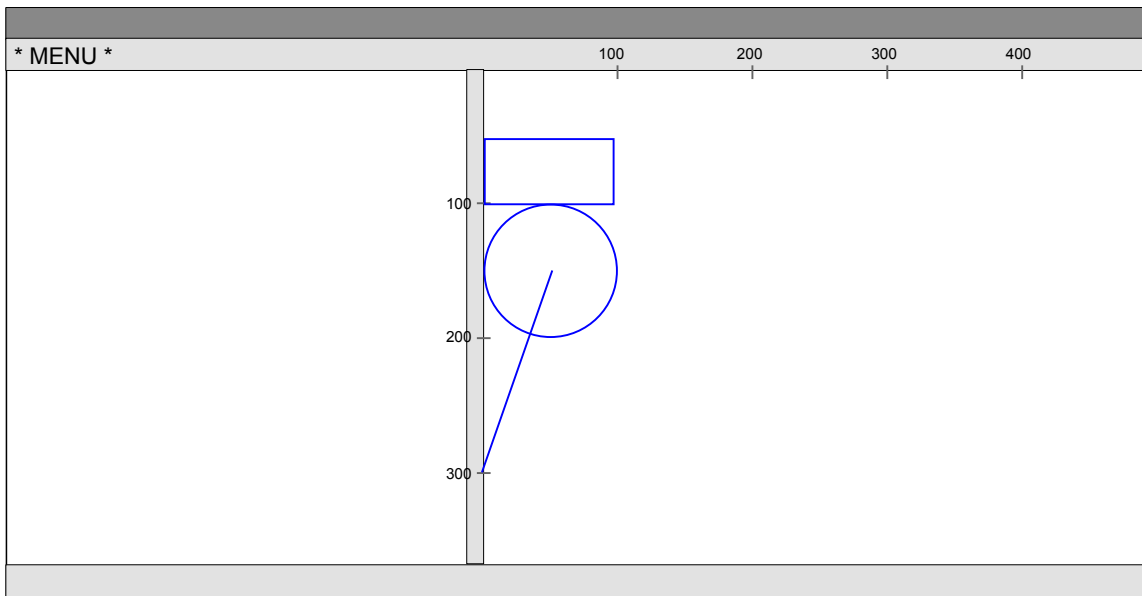
x: y: z:

```
Line 1: 6  
Line 2: 15  
Line 3: 18  
Line 4: 15
```

(Question 1 continued)**(b)** [4 marks] Understanding graphical output

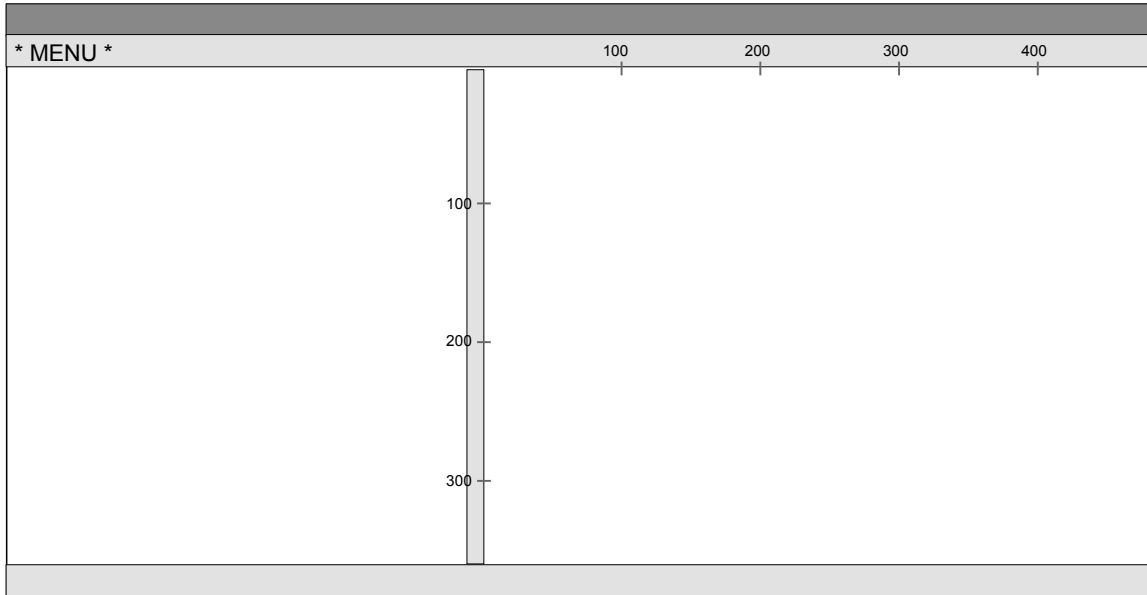
Sketch what the following method will draw in the graphics pane.

```
public static final double size = 100;  
public static final double dim = 50;  
public static final double mag = 150;  
  
public void drawIt(){  
    double x = dim - size/2;  
    double y = mag - size/2;  
    UI.drawOval(x, y, size, size);  
    UI.drawRect(x, y-size/2, size, dim);  
    UI.drawLine(dim, mag, 0, 300);  
}
```



(Question 1 continued)

Extra copy (in case you made a mistake):



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.

(c) [4 marks] Understanding if statement

The testIt method below has one parameter.

```
public void testIt (int x){  
    if(x > 10){  
        Ul.println ("first");  
    }  
    else if (x < 5){  
        Ul.println ("second");  
    }  
    else {  
        Ul.println ("third");  
    }  
  
    if (!(x==4 || x==0)){  
        Ul.println ("fourth");  
    }  
  
}
```

What would the following calls to testIt print out?

testIt(4); ⇒
second

testIt(8); ⇒
third
fourth

testIt(12); ⇒
first
fourth

Question 2. Writing programs with input, output and if

[9 marks]

Complete the `cabinBaggage` method below to check whether a carry on baggage meets the airline constraints. The method should ask the user for the length, the width, the height and the weight of the baggage, calculate the "linear dimension" of the baggage which is the sum of the length, the width and the height, and then print a message according to the following rules.

- If the "linear dimension" is greater than 118cm, print "too big"
- If the weight is greater than 7kg, print "too heavy"
- If it is not too big and it is not too heavy, print "all good"

For example, if you called `cabinBaggage` and entered 60 for the length, 50 for the width, 30 for the height, 8 for the weight, the text pane should print two messages:

```
too big
too heavy
```

If you called `cabinBaggage` and entered 20 for the length, 30 for the width, 30 for the height, 3 for the weight, the text pane should print one message:

```
all good
```



```

public void cabinBaggage() {
    Ul.clearText ();           // clears the text pane

    double length = Ul.askDouble("Enter Length: ");

    double width = Ul.askDouble("Enter Width: ");
    double height = Ul.askDouble("Enter Height: ");
    double weight = Ul.askDouble("Enter Weight: ");

    double linearDim = length + width + height;
    if (weight > 7){
        Ul.println ("too heavy");
    }
    if (linearDim > 118){
        Ul.println ("too big");
    }
    if ((weight <= 7) && (linearDim <= 118)){
        Ul.println ("yes, all good");
    }
}

```

Alternative **if** statement

```

if ((weight > 7) || (linearDim > 118)){
    if (weight > 7){
        Ul.println ("too heavy");
    }
    if (linearDim > 118){
        Ul.println ("too big");
    }
}
else{
    Ul.println ("yes, all good");
}

```

Please note that there are other ways to **do this**. For example, you can use a *boolean* variable, then one of your *boolean* expressions is much shorter.

Question 3. Writing methods that use objects

[7 marks]

This question is about a small game with two mice looking for food on a 5X5 board. The board is shown on the facing page and two cells (row 2, column 4) and (row 3, column 5) contain food. The board is drawn by a `setUpBoard` method and you do not need to define this method. Your task is to use a predefined class `Mouse` to create two mouse objects, and call the right methods to direct them to look for food.

Complete the `animate` method on the facing page, which should first create two `Mouse` objects as follows

- a red mouse at row 1, column 1, facing East
- a blue mouse at row 5, column 1, facing South

And then call the right methods on the objects to allow the two mice to find and eat food.

The `Mouse` class has one constructor and six methods as shown below:

Constructor:

```
public Mouse(int row, int col, String dir, Color c)
// constructs a Mouse object, with the specified position : row number and column number,
// the specified direction the mouse is facing : one of "North", "South", "East" "West"
// the specified color : one of the standard colors
// The constructor will assign these specified values to the fields , and call the draw method
// to display the mouse at the specified position on the graphics pane.
```

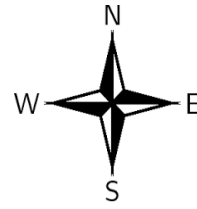
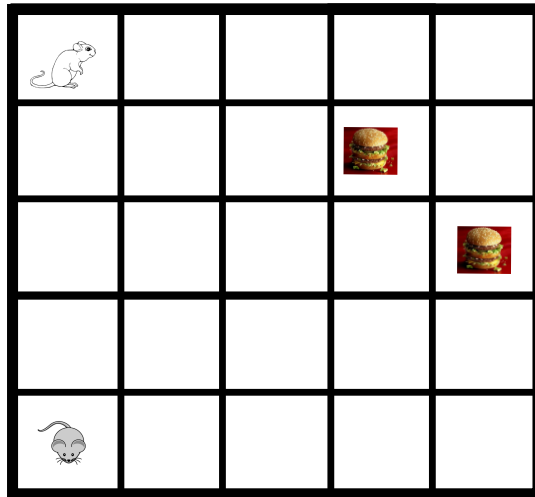
Methods:

```
public void move( int step)
// makes the Mouse move in the direction that it is facing for the specified number of steps
// The step can be 1 to 5, and is the number of rows or columns the mouse moves.
```

```
public void turn( String dir)
// changes the direction that the Mouse is facing ,
// Directions can be "North", "South", "East", or "West"
```

```
public void eat( )
// calls the eraseFood method to make the food disappear . It only works when the mouse is
// at the same position as the food.
```

The other three methods `draw()`, `eraseMouse()`, `eraseFood()` are **private** methods used only inside the `Mouse` class.



```

public void animate(){
    this.setUpBoard(); //draws the board and food. You do not need to define this method

    Mouse m1 = new Mouse(1,1, "East", Color.red);
    Mouse m2 = new Mouse(5,1, "South", Color.blue);
    m1.move(3);
    m1.turn("South");
    m1.move(1);
    m1.eat();
    m2.turn("North");
    m2.move(2);
    m2.turn("East");
    m2.move(4);
    m2.eat();

}

```

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. Understanding Arguments and Parameters

[7 marks]

The following printNote method calls the printOneLabel method several times. Show what printNote will print in the text pane.

```
public void printNote(){
    String name = "Tom";
    int x = 10;
    int y = 20;
    this.printOneLabel("Jack", x, y);
    this.printOneLabel(name, y, x);
    name = "Jane";
    this.printOneLabel(name, 5, x+1);
}

public void printOneLabel(String name, int x, int y){
    x = x + 1;
    UI.println (name + " @ " + x + ", " + y);
    UI.println ("---");
}
```

Write your answers here:

Jack @ 11, 20

Tom @ 21, 10

Jane @ 6, 11

Question 5. Writing methods with if and while

[10 marks]

This question is about a program that asks a user to guess a number. You are required to write the program using two methods. The first method will ask the user to guess one round. The second method will call the first method multiple times.

(a) [5 marks] Complete the first method `oneGuess` which asks the user for a number, and checks whether it matches the target number given in the parameter.

- If the guessed number matches the target, return true.
- If the guessed number is smaller than the target, print "too small", return false.
- if the guessed number is bigger than the target number, print "too big", return false.

```
public boolean oneGuess(int target){
    int input = Ul.askInt("Guess a number (1-10): ");
    if( input < target){
        Ul.println("too small");
        return false;
    }
    else if (input > target){
        Ul.println("too big");
        return false;
    }
    else{
        return true;
    }
}
```

(Question 5 continued)

(b) [5 marks] Complete the second method `guessGame` which first assign a random number to the variable `target` (code is giving), then calls the `oneGuess` method to allow the user to start guessing.

The user can guess at most five times. This method should stop once the user guessed correctly or the user has guessed five times. In the first case, it should print a "you win" message, and in the second case, it should print a "you lose" message. In both cases, the method should print the target number and the number of times the user has guessed.

For example, if the target number is 4, the user guessed 5, 2, 4, the text pane should look like this:

```
Guess a number(1-10): 5
too big
Guess a number(1-10): 2
too small
Guess a number(1-10): 4
You win!
The number is 4
You have guessed 3 times
```

```
public void guessGame(){
    //assign a random integer number between 1-10 to the variable target
    int target = (int) (Math.random() * 10 + 1);

    int count = 0;

    while(true){
        count++;
        if (this.oneGuess(target)){
            UI.println("You win");
            break;
        }else if (count>=5){
            UI.println("You lose");
            break;
        }
    }
    UI.println("The number is " + target);
    UI.println("You have guessed " + count + " times");
}
```

Student ID:
