

Family Name:

Other Names:

ID Number:

COMP 102: Test 2 | Model Solutions

8 May, 2009

Instructions

- Time allowed: **45 minutes**
- There are 45 marks in total.
- Answer **all** the questions.
- Write your answers in the boxes in this test paper and hand in all sheets. You may ask for additional paper if you need it.
- If you think some question is unclear, ask for clarification.
- Brief Java documentation will be supplied with the test.
- This test will contribute 15% of your final grade, 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

[14]

2. Two Class programs

[7]

3. Event Driven Input

[9]

4. Debugging

[15]

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. Basic Java

[14 marks]

(a) [4 marks] Consider the following printChoice method

```
public void printChoice(int x){
    if (x < 10) {
        System.out.println("Choice 1: " + x);
        x = x * 2;
    }
    if (x < 15) {
        System.out.println("Choice 2: " + x);
        x = x +100;
    }
    else if (x >= 15){
        System.out.println("Choice 3: " + x);
        x = 0;
    }
    System.out.println("x is now: " + x);
}
```

(i) [1 mark] What will be printed if printChoice(24) is called?

```
Choice 3: 24
x is now: 0
```

(ii) [1 mark] What will be printed if printChoice(10) is called?

```
Choice 2: 10
x is now: 110
```

(iii) [2 marks] What will be printed if printChoice(6) is called?

```
Choice 1: 6
Choice 2: 12
x is now: 112
.
```

(Question 1 continued on next page)

(Question 1 continued)

(b) [2 marks] What will the following fragment of Java print?

```
int i = 40;
System.out.print("answer: ");
while ( i > 1) {
    i = i / 2;
    System.out.print(i + ", ");
}
```

answer: | 20, 10, 5, 2, 1

(c) [4 marks] Write a fragment of Java that will print out each number from 1 to 500 (inclusive) followed by odd or even, like the following:

```
1 odd
2 even
3 odd
4 even
:
```

```
int k = 1;
while ( k<=500 ){
    System.out.println(k + " odd");
    System.out.println((k+1) + " even");
    k = k+2;
}
or for( int k=1; k<=500; k++){
    if (k%2==0) {
        System.out.println(k + " even");
    } else {
        System.out.println(k + " odd");
    }
}
```

(Question 1 continued on next page)

(Question 1 continued)**(d)** [4 marks] What will the following fragment of Java print?

```
int[ ] data = new int[10];

for (int i=0; i<data.length; i++){
    data[i] = i*i;
}
int index = 1;
while (index < 10) {
    System.out.println(index + " -> " + data[index]);
    index = index + 3;
}
```

```
1 -> 1
4 -> 16
7 -> 49
```

Question 2. Two Class programs

[7 marks]

Consider the PeekABoo class on the facing page that represents an image on the screen that can be "hidden" or visible.

Complete the following playPeekABoo method so that it uses the PeekABoo class to put two images on the screen, `rose.jpg` and `daisy.jpg`, at positions (10,10) and (300,10) respectively, and then repeatedly make each of them hidden, and then each of them visible again. The images should be hidden and shown 50 times (*i.e.*, a total of 100 actions on each image).

```
public void playPeekABoo (){
    JFrame frame = new JFrame("PeekABoo");
    DrawingCanvas canvas = new DrawingCanvas();
    frame.setSize(600, 400);
    frame.getContentPane().add(canvas, BorderLayout.CENTER);
    frame.setVisible(true);

    PeekABoo pb1 = new PeekABoo("rose.jpg", 10, 10, canvas);
    PeekABoo pb2 = new PeekABoo("daisy.jpg", 300, 10, canvas);
    int count = 0;
    while (count<50){
        pb1.hide();
        pb2.hide();
        pb1.unhide();
        pb2.unhide();
        count++;
    }
}
```

(Question 2 continued on next page)

(Question 2 continued)

```
import comp100.*;
```

```
public class PeekABoo{
    private DrawingCanvas canvas;
    private String imageName; // the name of the image file
    private int x;
    private int y;
    private boolean hidden = false;

    /** Construct a new PeekABoo object and draw it on the canvas. */
    public PeekABoo(String img, int x, int y, DrawingCanvas c){
        this.imageName = img;
        this.canvas = c;
        this.x = x;
        this.y = y;
        this.draw();
    }
    /** Sets the image to be hidden and redraws */
    public void hide(){
        this.hidden = true;
        this.draw();
    }
    /** Sets the image to be visible and redraws */
    public void unhide(){
        this.hidden = false;
        this.draw();
    }
    /** Draws the image (or a solid rectangle) on the canvas, and pauses */
    public void draw(){
        if (this.hidden){
            this.canvas.fillRect(this.x, this.y, 200, 200);
        }
        else{
            this.canvas.drawImage(this.imageName, this.x, this.y, 200, 200);
        }
        try{Thread.sleep(500);}catch(Exception e){}
    }
}
```

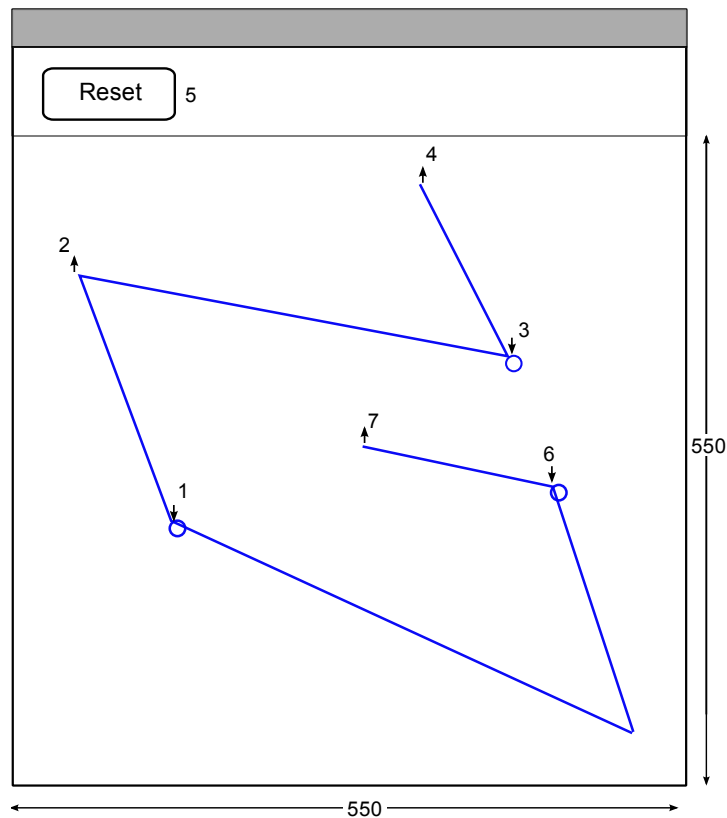
Question 3. Event-Driven Input

[9 marks]

(a) [6 marks] Consider the `TestEvents` class on the facing page, which constructs a simple GUI with one button and a canvas that responds to the mouse.

Sketch below what the program would draw on the canvas if the user took the following actions in sequence:

1. press mouse at point 1
2. release mouse at point 2
3. press mouse at point 3
4. release mouse at point 4
5. click on the Reset Button
6. press mouse at point 6
7. release mouse at point 7



(b) [3 marks] Modify the `TestEvents` class so that it has an additional button called "Clear" and pushing the "Clear" button will clear the canvas.

(Answer by modifying the code on the facing page)

(Question 3 continued on next page)

(Question 3 continued)

```

public class TestEvents implements ActionListener, MouseListener{
    private JFrame frame;
    private DrawingCanvas canvas;
    private int lastX = 500;
    private int lastY = 500;

    public TestGui(){
        this.frame = new JFrame("TestGui");
        this.frame.setSize(600, 600);
        this.canvas = new DrawingCanvas();
        this.canvas.addMouseListener(this);
        this.frame.getContentPane().add(this.canvas, BorderLayout.CENTER);
        JPanel panel = new JPanel();
        this.frame.getContentPane().add(panel, BorderLayout.NORTH);
        JButton button = new JButton("Reset");
        button.addActionListener(this);
        panel.add(button);
        -----> JButton cButton = new JButton("Clear");
        this.frame.setVisible(true);          cButton.addActionListener(this);
        panel.add(cButton);
    }
    public void actionPerformed(ActionEvent e){
        if (e.getActionCommand().equals("Reset")){
            lastX = 500;
            lastY = 500;
        }
        -----> else if (e.getActionCommand().equals("Clear")){
            this.canvas.clear();
        }
    }
    public void mousePressed(MouseEvent e) {
        int x = e.getX();
        int y = e.getY();
        this.canvas.drawOval(x, y, 5, 5);
        this.canvas.drawLine(this.lastX, this.lastY, x, y);
        lastX = x;
        lastY = y;
    }
    public void mouseReleased(MouseEvent e) {
        int x = e.getX();
        int y = e.getY();
        this.canvas.drawLine(this.lastX, this.lastY, x, y);
        lastX = x;
        lastY = y;
    }
    public void mouseClicked(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {} public void mouseExited(MouseEvent e) {}
}

```

Question 4. Debugging

[15 marks]

The `breakLines` method is intended to read a file containing words, and print them out, separated by spaces, ensuring that there are at most three words on each line. If there are any numbers among the words, then it will not print the numbers, but ensure that a word immediately after a number is at the beginning of a line.

For example, if the file `test.txt` contains the following text:

```
ant bee cat dog emu 2 fox gnu hen 8 jay kea owl pig
```

then `breakLines("test.txt")` should print out:

```
ant bee cat
dog emu
fox gnu hen
jay kea owl
pig
```

The following version of `breakLines` has errors:

```
public void breakLines(String fname){
    try {
        Scanner scan = new Scanner(new File(fname));
        int wordCount = 0;
        while (scan.hasNext()) {
            String word = scan.next();
            if ( scan.hasNextInt() ){
                System.out.print(word+"\bksln"); //print word and start new line
            }
            if ( wordCount == 3 ){
                System.out.print(word+"\bksln");
                wordCount = 0;
            }
            else {
                System.out.print(word);
                wordCount++;
            }
        }
        scan.close();
    } catch (IOException e){System.out.println("Error while scanning"+e);}
}
```

(a) [7 marks] What would the statement `breakLines("test.txt");` print out?

```
antbeecatdog
emu
emu2foxgnu
hen
hen8jaykea
owlpig
```

(Question 4 continued)**(b)** [8 marks] Write a correct version of the breakLines method.

```

public void breakLines(String fname){
    try {
        Scanner scan = new Scanner(new File(fname));
        int wordCount = 0;
        while (scan.hasNext()) {
            String word = scan.next();
            while ( scan.hasNextInt() ){
                wordCount = 3; // assert that this is the last word on the line
                scan.next(); // throw away the number
            }
            if ( wordCount >= 2 ){
                System.out.println(word); // print word and a new line
                wordCount = 0;
            }
            else {
                System.out.print(word+ " ");
                wordCount++;
            }
        }
    }
}

```

OR -----

```

        if ( scan.hasNextInt() ){ //throw away a number
            scan.next();
            if ( wordCount > 0 ){ System.out.println();} // start a new line
            wordCount = 0;
        }
        else if ( wordCount == 3 ){ // we have done three words:
            System.out.println (); // start a new line
            wordCount = 0;
        }
        else { // safe to print the word on this line
            String word = scan.next();
            System.out.print(word+ " ");
            wordCount++;
        }
    }
}

```

```

        scan.close();
    } catch(Exception e){System.out.println("Error while scanning"+e);}
}

```
