

University of Cape Town
 Department of Computer Science
 Computer Science CSC116S

Test 2 Solutions - 14 September 2005

- Answer all questions.
- All questions that refer to elements of programming make reference to the Java programming language as studied in class.
- Good luck !

Marks: 30

- Approximate marks per question are shown in brackets

Time: 40 minutes

- The use of calculators is permitted

	Surname		Initials
NAME:			

STUDENT NO:			COURSE CODE:	CSC
--------------------	--	--	---------------------	-----

This paper consists of 3 questions and 7 pages (including this cover page).

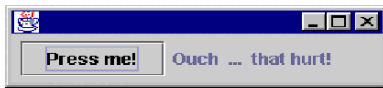
Mark Allocation							
Quest	Marks	Internal	External	Quest	Marks	Internal	External
1	[12]			3	[8]		
2	[10]						
Total				Total			
Grand Total							
Final Mark							
Internal Examiner:				External Examiner:			

Section 1. Graphical User Interfaces

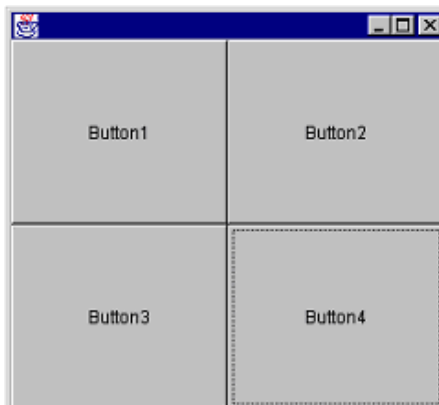
Question 1. [12 marks]

a) Draw the component hierarchy diagram for the figure below.

[1]



b) Using GridLayout Manager please write the Java code to implement the GUI below.



```

import java.awt.*; public class TestGrid extends Frame {
    public static void main(String args[]) {
        Panel p = new Panel();
        Button b1 = new Button("Button1");
        Button b2 = new Button("Button2");
        Button b3 = new Button("Button3");
        Button b4 = new Button("Button4");
        TestGrid t = new TestGrid();
        t.setSize(150,150);
        t.setVisible(true);
        t.add(p);
        p.setLayout(new GridLayout(2,2));
        p.add(b1 );
        p.add(b2);
        p.add(b3);
        p.add(b4);
    }
}

```

[4]

- c) Look at the code below. Extend the program so that the line “ok, now I’m really pressed.” appears on the screen each time the button is pressed.

```

public class ButtonTest extends Frame {
    public static void main(String args[]) {
        ButtonTest bt = new ButtonTest();
    }

    public ButtonTest() {
        JButton b = new JButton("Test");
        add(b);
        pack();
        setVisible(true);
    }

    public void ButtonPressed(Button b) {
        System.out.println("pressed !");
    }
}

```

Hint: listener, event handling.

Solution:

extensions contain:

implements ActionListener..., [1 mark]

in Constructor: b.addActionListener(this); [1 mark]

and the method ButtonPressed(Button b) is renamed to
public void ActionPerformed(ActionEvent e) [1 mark]

[3]

d) Which of the following declarations are correct?

1. actionPerformed(ActionEvent e) {}
2. mousePressed(MouseEvent e) {}
3. itemStateChanged(ItemAdded e) {}
4. All of the above

Soln : 1 and 2

[2]

e) What are layout managers used for in java? Give an example.

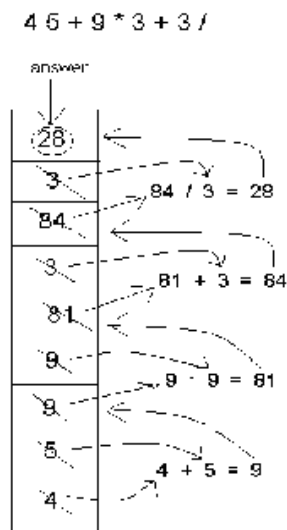
[2]

Section 2. Stacks, Queues and Trees

Question 2. [10 marks]

- a) Calculate the value of the following Reverse Polish Notation expression using a Stack. Remember to show all your working

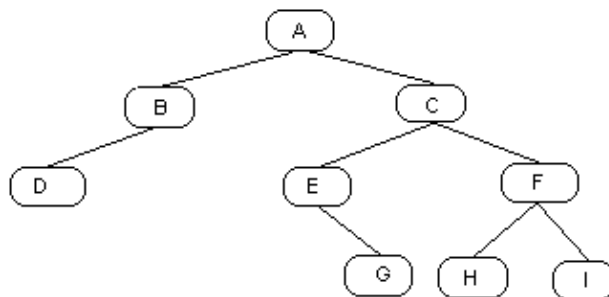
$$45+9*3+3/$$



- 1) Push 4 onto the Stack (1 item in Stack)
- 2) Push 5 onto the Stack (2 items in Stack)
- 3) See the +, so pop 2 operands off (4 and 5) (2 items in Stack)
- 4) Push their result (9) onto the Stack (1 item in Stack)
- 5) Push 9 onto the Stack (2 items in Stack)
- 6) See the *, so pop 2 operands off (9 and 9) (1 item in Stack)
- 7) Push their result (81) onto the Stack (1 item in Stack)
- 8) Push 3 onto the Stack (2 items in Stack)
- 9) See the +, so pop 2 operands off (81 and 3) (1 item in Stack)
- 10) Push their result (84) onto the Stack (1 item in Stack)
- 11) Push 3 onto the Stack (2 items in Stack)
- 12) See the /, so pop 2 operands off (84 and 3) (1 item in Stack)
- 13) Push their result (28) onto the Stack (1 item in Stack)
- 14) See that there is nothing left in the expression, so pop the final element (28) off of the Stack, and you know that must be your answer. (0 items in Stack)

[2]

- b) Please traverse the tree in the figure below using Preorder, Postorder and Inorder traversals.



Preorder

Soln : A B D C E G F H I

[2]

Postorder

Soln : D B G E H I F C A

[2]

Inorder

Soln : D B A E G C H F I

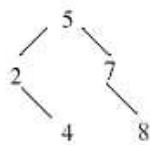
[2]

- c) Given an empty binary search tree of integers, show the structure of the tree after each of the values 5, 2, 4, 7, 8, 1, 3 is inserted. Show the steps not just the final solution.

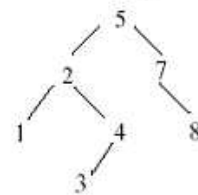
Solution:
After inserting 5,2,4



After inserting 7, 8



After inserting 1, 3



[2]

Section 3. Graphics in Java

Question 3. [8 marks]

For the following questions, assume the Java2D graphics primitives:

Arc2D.Float (x, y, width, height, start, extent, type)

Ellipse2D.Float (x, y, width, height)

Line2D.Float (x1, y1, x2, y2)

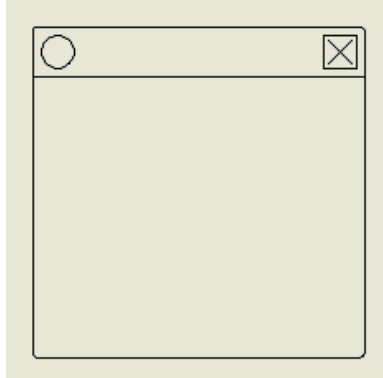
Rectangle2D.Float (x, y, width, height)

RoundRectangle2D.Float (x, y, width, height, arcwidth, archeight)

- a) Draw the figure that results from the following sequence of drawing statements in Java2D.

```
canvas.draw (new RoundRectangle2D.Float (100, 100, 200, 200, 5, 5));
canvas.draw (new Line2D.Float (100, 130, 300, 130));
canvas.draw (new Rectangle2D.Float (275, 105, 20, 20));
canvas.draw (new Line2D.Float (278, 108, 292, 122));
canvas.draw (new Line2D.Float (278, 122, 292, 108));
canvas.draw (new Ellipse2D.Float (105, 105, 20, 20));
```

Solution : 1 mark for each correct item.



[6]

- b) Why must we use the paintComponent method to draw graphics instead of drawing them from the containing Frame's constructor?

so that Java will be able to redraw the component whenever necessary e.g., when the window is minimised and restored. (1 for reason, 1 for example) [2]