**King Fahd University of Petroleum & Minerals**

**College of Computer Science and Engineering**

**Information and Computer Science Department**

**ICS 201 – Introduction to Computing II**

**Summer Semester 2011-2012 (113)**

**SOLUTION to Major Exam 02**

**15th July 2012**

**Time: 120 minutes**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ StudentID: \_\_\_\_\_\_\_\_\_\_\_**

This exam consists of four questions. All questions must be answered.

|  |  |  |
| --- | --- | --- |
| **Question#** | **Max Marks** | **Marks Obtained** |
| 1 | 20 |  |
| 2 | 20 |  |
| 3 | 30 |  |
| 4 | 30 |  |
| Total | 100 |  |

Q. 1 [20 marks] For each of the following statements, select one of the four choices A, B, C or D.

1. For the following code snippet, which of the following statements is true:

**int x = 5, y = 0;**

**double z = x/y;**

**System.out.println(z);**

1. The code generates an exception during compile-time
2. ***The code generates an exception during run-time***
3. The code does not generate an exception
4. The output of the program is NaN (Not a Number)
5. The program generates a DivisionByZeroException

2. For the following code snippet, which of the following statements is true:

**String[] x = new String[3];**

**x[0] = "la"; x[1] = null; x[2] = "55";**

**for(int i = 0; i < x.length; i++)**

**try {**

**String y = x[i].substring(1);**

**System.out.println(Integer.parseInt(y));**

**}**

**catch(NullPointerException e1) {**

**System.out.println("It’s a NullPointerException");**

**}**

**catch(NumberFormatException e2) {**

**System.out.println("It’s a NumberFormatException");**

**}**

1. This code generates (and handles) only a NumberFormatException
2. This code generates (and handles) only a NullPointerException
3. The code does not generate any exception
4. ***This code first generates (and handles) a NumberFormatException and then a NullPointerException***
5. This code first generates (and handles) a NullPointerException and then a NumberFormatException

3. Which of the following statements is true:

1. ***In any one execution of a try block, at most one exception can be thrown.***
2. In any one execution of a try block, more than one exception can be thrown
3. In any one execution of a try block, at least one exception can be caught
4. In any one execution of a try block, more than one exception can be caught
5. None of the above

4. For the following code snippet, which of the following statements is true:

**JLabel l1;**

**try {**

**Thread.sleep(1000);**

**l1.setText("Remaining Time: 03:00");**

**}**

**catch(InterruptedException e) {**

**//Don't do anything**

**}**

**finally {**

**System.out.println("This is always executed");**

**}**

1. ***This code generates a NullPointerException and then the program exits***
2. This code generates an InterruptedException and then the program exits
3. This code prints “This is always executed” and then the program exits.
4. This code first prints “This is always executed”, then generates an exception.
5. This code first generates an exception, then prints “This is always executed”.

5. For the method **setDefaultCloseOperation(int operation)** for the class **JFrame**, the default operation is (i.e. if you don’t specify any operation)

1. **JFrame.EXIT\_ON\_CLOSE**
2. **JFrame.DISPOSE\_ON\_CLOSE**
3. ***JFrame.HIDE\_ON\_CLOSE***
4. **JFrame.DO\_NOTHING\_ON\_CLOSE**
5. **JFrame.CLEAR\_ON\_CLOSE**

6. What happens if a programmer forgets to associate his component with a listener (for example, he forgets to write **button.addActionListener(this);** ?

1. The program will not compile
2. The program will compile, but generate a run-time exception
3. ***The program will compile and run, but the component button will not respond to clicks***
4. The program will compile and run and the component button will respond to clicks
5. None of the above

7. Which of the following statement is false:

1. A **GridLayout** stretches the components to the size of the enclosing container.
2. A **BorderLayout** stretches the components to the size of the enclosing container.
3. A **FlowLayout** does not stretch the components to the size of the enclosing container.
4. ***In a JFrame of size 400x400 with a BorderLayout, the size of the EAST panel is greater than the size of the NORTH Panel.***
5. In a JFrame of width 400x400 with a **BorderLayout**, the size of the SOUTH panel is greater than the size of the WEST Panel.

8. Which of the following statements is true:

1. An applet contains a **setSize(int, int)** method.
2. An applet contains a **setDefaultCloseOperation(int operation)** method.
3. An applet contains a **setVisible(boolean)** method.
4. ***An applet contains a setLayout(LayoutManager) method.***
5. An applet contains a **main** method.

9. Which of the following lines of code will NOT draw a circle.

1. g.drawOval(100, 100, 200, 200);
2. g.fillOval(100, 100, 200, 200);
3. g.drawArc(100, 100, 200, 200, 180, 360);
4. ***g.fillArc(100, 100, 200, 200, 360, 180);***
5. g.drawRoundRect(100, 100, 200, 200, 200, 200);

10. Which of the following method invocations will make a Thread sleep for 0.5 seconds:

1. Thread.sleep(5);
2. Thread.sleep(50);
3. ***Thread.sleep(500);***
4. Thread.sleep(5000);
5. Thread.sleep(50000);

Write your answers here:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 B | 2 D | 3 A | 4 A | 5 C |
| 6 C | 7 D | 8 D | 9 D | 10 C |

Q. 2 [**7+7+6 = 20 marks**] For each of the following programs, find the output:

|  |
| --- |
| **import java.awt.\*; import javax.swing.\*; import java.awt.event.\*;**  **public class ArcsExample extends JFrame {**  **public ArcsExample() {**  **super("Eating Pac Men");**  **setSize(300, 300);**  **setVisible(true);**  **}**    **public void paint(Graphics g) {**  **super.paint(g);**  **g.setColor(Color.BLACK);**  **g.fillArc(220, 140, 20, 20, 30, 300);**    **g.setColor(Color.BLUE);**  **g.fillArc(180, 120, 60, 60, 30, 300);**    **g.setColor(Color.RED);**  **g.fillArc(50, 50, 200, 200, 30, 300);**  **}**    **public static void main(String[] args) {**  **ArcsExample a = new ArcsExample();**  **}**  **}** |
| **import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;**  **public class LoginFrameApplet extends JApplet {**  **JLabel uL, pL; JTextField uT, pT; JButton oB, cB;**  **JPanel dataP, buttonP;**  **JPanel dummyPanel1, dummyPanel2, dummyPanel3;**    **public void init() {**  **uL = new JLabel("User Name"); pL = new JLabel("Password");**  **uT = new JTextField(20); pT = new JTextField(20);**  **oB = new JButton("Ok"); cB = new JButton("Cancel");**  **dataP = new JPanel(); dataP.setLayout(new GridLayout(2, 2));**  **dataP.add(uL); dataP.add(uT); dataP.add(pL); dataP.add(pT);**  **buttonP = new JPanel(); buttonP.setLayout(new FlowLayout());**  **buttonP.add(oB); buttonP.add(cB);**  **dummyPanel1 = new JPanel(); dummyPanel2 = new JPanel();**  **dummyPanel3 = new JPanel(); setLayout(new BorderLayout());**  **add(dataP, BorderLayout.CENTER);**  **add(buttonP, BorderLayout.SOUTH);**  **add(dummyPanel1, BorderLayout.WEST);**  **add(dummyPanel2, BorderLayout.EAST);**  **add(dummyPanel3, BorderLayout.NORTH);**  **}**  **}** |
|  |

|  |
| --- |
| **class MyThread extends Thread {**  **private int time = 120;**  **private int step;**  **private String name;**  **public MyThread(String name, int step) {**  **this.name = name;**  **this.step = step;**  **}**    **public void run() {**  **do**  **{**  **try {**  **Thread.sleep(1000);**  **}**  **catch(InterruptedException e) {}**  **time = time - step;**  **System.out.println(name + ": " + time);**  **} while(time > 0);**  **}**  **}**  **public class ThreadTest {**  **public static void main(String[] args) {**  **MyThread t1 = new MyThread("Slow", 30);**  **MyThread t2 = new MyThread("Medium", 40);**  **MyThread t3 = new MyThread("Fast", 60);**    **t1.start(); t2.start(); t3.start();**  **}**  **}** |
| **Slow: 90**  **Medium: 80**  **Fast: 60**  **Fast: 0**  **Medium: 40**  **Slow: 60**  **Medium: 0**  **Slow: 30**  **Slow: 0** |

Q. 3: [25 + 5 = 30 marks] Write an applet in Java that draws concentric circles as shown below:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

The bounding box of the first circle is has top-left coord = (0, 0) and (width, height) = (100, 100).

The thickness of each ring is 20 pixels. There is a time lapse of 1 second between the drawing of each circle.

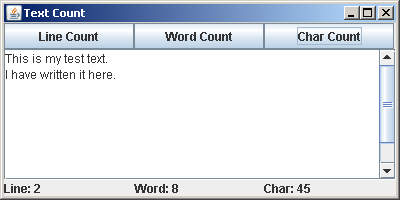
1. Write HTML code for running this applet:

**<applet code="ConcentricCircles.class" width=200 height=200></applet>**

1. [25 marks] Write the code for the applet.

|  |
| --- |
| import java.awt.\*;  import javax.swing.\*;  import java.awt.event.\*;  public class ConcentricCircles extends JApplet implements Runnable {    int circle;  public void init() {  circle = 0;  Thread t = new Thread(this);  t.start();  }  public void paint(Graphics g) {  //super.paint(g);  int val = 100, step = 20, size;  circle = (circle + 1) % 6;  if(circle % 2 == 0)  g.setColor(Color.YELLOW);  else  g.setColor(Color.white);  g.fillOval(val + circle\*step - 100, val + circle\*step - 100,  2\*(val - circle\*step), 2\*(val - circle\*step));    if(circle % 6 == 5) {  g.setColor(getBackground());  g.fillRect(0, 0, this.getWidth(), this.getHeight());  }  }      public void run() {  while(true) {  try {  Thread.sleep(1000);  repaint();  }  catch(InterruptedException e) {}  }  }  } |

Q. 4: [25 + 5 = 30 marks] Write a program in Java that constructs the following GUI:



Each of the buttons Line Count, Word Count and Char Count count the number of lines, words and characters in the text entered in the text area. Assume that two lines are separated by a ‘\n’, two words are separated by a ‘ ‘ (space).

1. [15 marks] Part of the program is given here. Write code for the constructor only:

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class WordCount extends JFrame implements ActionListener {

private JButton line, word, chac;

private JTextArea jta;

private JScrollPane js;

private JLabel ln, wd, cc;

private JPanel bPanel, tPanel, lPanel;

public WordCount() {

super("Text Count");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

line = new JButton("Line Count");

word = new JButton("Word Count");

chac = new JButton("Char Count");

jta = new JTextArea(10, 10);

js = new JScrollPane(jta);

ln = new JLabel("Line: ");

wd = new JLabel("Word: ");

cc = new JLabel("Char: ");

bPanel = new JPanel();

lPanel = new JPanel();

setLayout(new BorderLayout());

bPanel.setLayout(new GridLayout(1, 3));

lPanel.setLayout(new GridLayout(1, 3));

bPanel.add(line);

bPanel.add(word);

bPanel.add(chac);

add(bPanel, BorderLayout.NORTH);

add(js, BorderLayout.CENTER);

lPanel.add(ln);

lPanel.add(wd);

lPanel.add(cc);

this.add(lPanel, BorderLayout.SOUTH);

line.addActionListener(this);

word.addActionListener(this);

chac.addActionListener(this);

setSize(400, 200);

setVisible(true);

}

public static void main(String[] args) {

WordCount wc = new WordCount();

}

public void actionPerformed(ActionEvent ae) {

String text = jta.getText();

int lineval = 0, wordval = 0, charVal = 0, pointer = 0;

while(pointer < text.length()) {

if(text.charAt(pointer) == '\n')

lineval++;

if(text.charAt(pointer) == ' ')

wordval++;

charVal = pointer;

pointer++;

}

wordval = wordval + lineval;

//Optional, for correct input

if(ae.getSource() == line)

ln.setText("Line: "+lineval);

if(ae.getSource() == word)

wd.setText("Word: "+wordval);

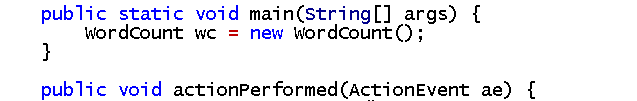
if(ae.getSource() == chac)

cc.setText("Char: "+charVal);

}

}

1. [15 marks] Write code for the ActionListener



Answer included in part (a) [See above]