Review for Final
Pic 20

1. Write a (syntactically correct) method that will take a string, replace all the o's in it by a, replace all the e's by a, and leave all other symbols in the string alone. Thus if it would chance "love" to "lava". (Assume all letters are lower case)

2. Write a method that input an integer between 0 and 51 will return a string that is the name of a card from a standard deck. The rules are:

- a integer between 0 and 12 is to be a club
- a integer between 13 and 25 is to be a diamond
- a integer between 26 and 38 is to be a spade
- a integer between 39 and 51 is to be a heart

- a integer congruent 12 mod 13 is to be an ace
- a integer congruent 11 mod 13 is to be a king
- a 10 mod 13 is to be a queen
- a 9 mod 13 is to be a jack
- a 8 mod 13 is to be a 10
etc.

Thus, for example, 38 will correspond to the ace of spades; 0 will correspond to the 2 of clubs.

3. Write a class, to be called SumOfPowers that will have two constructors. The first will be used to compute sums of the form

\[ 1^k + 2^k + 3^k + \ldots + n^k, \]

the sum of the \( k \)th powers of the integers from 1 to \( n \), where \( k \) is an integer and \( n \) is a positive integer. The second constructor is to compute sums of the form

\[ a^k + (a+1)^k + (a+2)^k + \ldots + b^k, \]

the sum of the \( k \)th powers of all the integers between \( a \) and \( b \), where \( k \) is an integer and \( a \) and \( b \) are positive integers with \( a \leq b \).

All the variables of the class are private, and it is to contain a getSum() method.

4. Write a driver application for the class in problem 3, that will print out (via System.out.println()) the two sums

\[ 1^3 + 2^3 + 3^3 + \ldots + 10^3 \]

and

\[ 50^2 + 51^2 + \ldots + 100^2 \]
5. What problems might be foreseen if you used the methods of problems 3 and 4 to calculate the sum

\[ \sum_{i=1}^{100} i^5 \]

6. Discuss the methods you would employ to solve the card shuffling problem: A deck of 52 cards is to be shuffled and dealt out to four hands. Each hand (one north, one east, one south, and one west) is to receive 13 cards. The cards in each hand are to be shown face-up, in a 13x1 grid with clubs coming first, diamonds second, spades, third, and finally hearts. Within each suit, the cards in each suit are to be displayed in increasing order as you go down the grid.

You may assume that a permutation of the 52 integers from 0 to 51 represents a shuffle of the deck.

7. The following program just mimics input:

```java
public class eee {
    public String Itself(String st)
    {
        return st;
    }

    public static void main(String args[])
    {
        eee app = new eee();
        String test = "My country tis of thee";
        String result= app.Itself(test);
        System.out.println("\n\n" + test);
        System.exit(0);
    }
}
```

Modify it so that if the input string has three or more lower case e's it will throw an exception. That is, if the string entered is changed to `String test = "My country tis of theee";` it will print out

```java
Too many e's in My country tis of theee
```
8. You are too write an applet that provides a drill in multiplication for any two one digit numbers, each of which is between 2 and 9. When the applet first opens the screen will look like this:

When the user clicks on the "Next" button the next screen will present a multiplication problem in the first text field:

In addition the second text field, the one that had "Click Next -> to start", will be cleared.

Continuing, the user will type in the answer and press enter. If the right answer is entered the "right, the answer is 45" will appear in the third text field.

The user continues by pressing on the Next button. When this is done a new problem appears in the first field, and the second and third fields are cleared.

If the answer entered is wrong the applet will say so:
The user then continues by pressing the Next button.

Specifications: The first three elements of the 1 X 4 grid are text fields. The first and third are set ineditable.

You may assume that the entry in the second field is syntactically correct; that is it the Integer.parseInt() method will return an integer when applied to the second field entry.

Your program must clear the second and third fields when the Next button is cleared. When he is done, he exits by pressing the window closing icon.

An outline of the program I wrote is listed below. The init() block is correct. Your problem is to complete the actionPerformed(ActionEvent e) and setProblem() methods.

```java
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class Multiplication extends JApplet implements ActionListener {
    private JTextField textField1, textField2, textField3;
    private JButton next;
    String problem;
    int product;

    public void init() {
        Container c = getContentPane();
        c.setLayout( new GridLayout(1,4) );
        textField1 = new JTextField();
        textField1.setEditable(false);
        c.add(textField1);

        textField2 = new JTextField();
        textField2.setText("Click Next -> to start");
        textField2.addActionListener(this);
        c.add(textField2);

        textField3 = new JTextField();
        textField3.setEditable(false);
        c.add(textField3);

        next = new JButton("Next");
        next.addActionListener(this);
        c.add(next);
    }
}
```
public void actionPerformed(ActionEvent e)
{
    if (e.getSource() == next)
    {
        problem = setProblem();
    }
    if (e.getSource() == textField2)
    {
        int n = Integer.parseInt(e.getActionCommand());
    }
}

public String setProblem()
{
    int a = 2 + (int)(Math.random())*8;
    int b = 2 + (int)(Math.random())*8;
    product = a*b;
}
}// end class Multiplication

9. Write a method

    public double evaluate(String st, double u)
    {
    }
}

where st is a string that has 5 elements

    3.567*x^7

where: the first element is a double (it need not be 3.567), x is a variable, and the last element is a positive integer (it need not be 7). The double u is any double number. The value the method is to return is the value you get when you substitute u for x and then calculate the numerical value of the expression.
10. A thread's `run()` method includes the following:

```java
1. try
2. {
3.     sleep(100)
4. }
5. catch (InterruptedException ie) {}
```

Assuming the thread is not interrupted, which of the following statements is correct?

(a) The code will not compile, because exceptions may not be caught in a thread's `run()` method.

(b) At line 3 the thread will stop running. Execution will resume in, at most, 100 milliseconds.

(c) At line 3 the thread will stop running. Execution will resume in exactly 100 milliseconds.

(d) At line 3 the thread will stop running. Execution will resume some time after 100 milliseconds.

11. True or False: If a thread has a `wait()` instruction in the `try` block of its `run()` method then it must also have a `notify()` instruction in its `run()` method. If false, why?

12. True or False: Once you create an instance of a thread in an application the `run()` method begins. If false, why?

13. The Painter.java application we did for the mouse permitted the user to make free hand drawings on the screen:

The code is on the next page.
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class Painter extends JFrame
{
    private int xValue = -10, yValue = -10;

    public Painter()
    {
        getContentPane().add
        {
            new JLabel("Drag the mouse to draw"),
            BorderLayout.SOUTH
        };

        addMouseMotionListener
        {
            new MouseMotionAdapter()
            {
                public void mouseDragged(MouseEvent e)
                {
                    xValue = e.getX();
                    yValue = e.getY();

                    repaint();
                }
            }
        };

        setSize(300,300);
        setLocation(300,300);
        show();
    } // end Painter()

    public void paint(Graphics g)
    {
        g.fillOval(xValue, yValue, 4, 4);
    }
}
public static void main(String args[]) {
    Painter app = new Painter();

    app.addWindowListener
    (new WindowAdapter()
    {
        public void windowClosing(WindowEvent e)
        {
            System.exit(0);
        }
    });

} // end main()

} // end class Painter

Modify the code so that the lower, right hand quadrant is forbidden territory. That is, if the mouse is dragged into the region x ≥ 150, y ≥ 150 the program will produce a message and then close. The output would be:

![](image)

14. The program above uses new MouseMotionAdapter(), as well as new WindowAdapter() What, in general, is an adapter? What is its' purpose?