

CSCE 145: Final Exam Key, Spring 2011

All questions are worth 10 points each.

1. Implement a class **Sphere** that has one instance variable of type double for the *radius* of the sphere. Do not include any getters or setters. However, implement the method *getSphereVolume()*, where the volume of a sphere is $\frac{4}{3}\pi r^3$ for radius *r*.

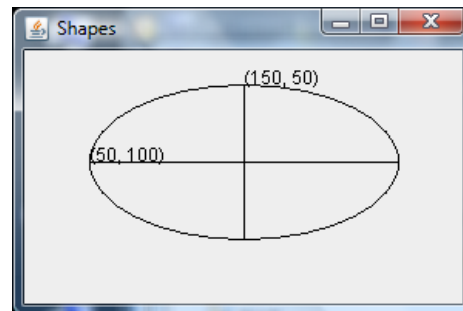
```
public class Sphere {
    private double radius;
    public double getSphereVolume() {
        return 4.0 * Math.PI * radius * radius * radius / 3.0;
    }
}
```

2. Write a static method called *swap* that has a String as a parameter. The method scrambles the String by generating two random numbers between 0 and the length of the String and then switching the characters located at these indexes. It then returns the resultant String. For example, if the method was invoked as *swap("computing")* and it generated the random numbers 2 and 3, then the String to be returned would be "copmting".

```
public static String swap(String s) {
    int index1 = (int)(Math.random()*s.length());
    int index2 = (int)(Math.random()*s.length());
    char c1 = s.charAt(index1);
    char c2 = s.charAt(index2);
    if (index1 < index2) {
        String s1 = s.substring(0, index1-1);
        String s2 = s.substring(index1+1, index2-1);
        String s3 = s.substring(index2+1);
    } else {
        String s1 = s.substring(0, index2-1);
        String s2 = s.substring(index2+1, index1-1);
        String s3 = s.substring(index1+1);
    }
    return s1+c1+s2+c2+s3;
}
```

3. Define the class **Shapes** that extends **JFrame** and produces the following graphical window consisting of an oval and two lines (you do not need to write the strings (150, 50) and (50, 150)):

```
public class Shapes extends JFrame {
    public Shapes() {
        super("Shapes");
        setSize(300, 200);
    }
    public void paint(Graphics g) {
        super.paint(g);
        g.setColor(Color.BLACK);
        g.drawOval(50, 50, 200, 100);
        g.drawLine(50, 100, 250, 100);
        g.drawLine(150, 50, 150, 150);
    }
    public static void main(String[] args) {
        Shapes s = new Shapes();
        s.setDefaultCloseOperation(EXIT_ON_CLOSE);
        s.setVisible(true);
    }
}
```



4. In the poison apple game, the computer offers 15 apples to three players. The last apple is a poisonous apple and whoever chooses this last apple loses the game. The players take turns removing either one, two, or three apples, with the goal being to leave exactly one apple that one of their opponents must then choose. Implement the main method for this game, using a loop to ask each

player in turn how many apples he/she chooses. The loop should end when there are no apples left. Use the **Scanner** class to get the number of apples from each player. Your program should print which player wins.

```
public static void main (String[] arg) {
    int applesLeft = 15;
    int playerNumber = 0;
    int applesChosen = 0;
    Scanner kb = new Scanner (System.in);
    while (applesLeft > 1) {
        System.out.print("Player " +playerNumber+
            ": how many apples 1, 2, 3 will you remove? ");
        applesChosen = kb.nextInt();
        applesLeft = applesLeft - applesChosen;
        playerNumber = (playerNumber + 1) % 3; // switch players
    }
    System.out.println("Winner is player number " + playerNumber);
}
```

5. Create a graphics program (a JApplet, which has a *paint(Graphics canvas)* method) that uses the method *drawString* to write on the applet the coordinates of the mouse when it is clicked. That is, whenever a user clicks the mouse, its coordinates will be written at that point in the applet window. Your class will need to extend **JApplet** and implement **MouseListener**.

```
public class MouseCoordinates extends JApplet implements MouseListener {
    private int mX, mY;
    public void init() {
        Container c = this.getContentPane();
        c.addMouseListener(this);
    }
    public void paint(Graphics g) {
        g.drawString("Mouse at "+mX+", "+mY, mX, mY);
    }
    public void mouseClicked(MouseEvent e) {
        mX = e.getX();
        mY = e.getY();
        repaint();
    }
    public void mouseEntered(MouseEvent arg0) {}
    public void mouseExited(MouseEvent arg0) {}
    public void mousePressed(MouseEvent arg0) {}
    public void mouseReleased(MouseEvent arg0) {}
}
```

6. Convert the following *for*-loop into a *while*-loop:

```
Scanner kbd = new Scanner(System.in);
for (String ask = kbd.nextLine(); ask != null; ask = kbd.nextLine()) {
    if (ask.contains("stop"))
        System.exit(0);
    else
        replyTo(ask);
}
Scanner kbd = new Scanner(System.in);
String ask = kbd.nextLine();
while (ask != null) {
    if (ask.contains("stop"))
        System.exit(0);
    else
        replyTo(ask);
}
```

```
ask = kbd.nextLine();
}
```

7. Write a program (just a main method) that tests the formula

$$\frac{\pi^2}{6} = \sum_{k=1}^n \frac{1}{k^2}$$

Your program should ask the user for a value for n , compute the value of the expression on each side of the equation, and then print n , those two values, and their difference.

```
public static void main(String[] args) {
    Scanner kbd = new Scanner(System.in);
    System.out.print("Please input a value for n: ");
    int n = kbd.nextInt();
    double left = Math.PI * Math.PI / 6.0;
    double right = 0;
    for (int k = 1; k <= n; k++)
        right = right + 1.0 / (k * k);
    double diff = left - right;
    System.out.println("n = " + n + " left = " + left + " right = " + right
        + "difference = " + diff);
}
```

8. Write a method named *reverseFile* that has one parameter, the name of a text file, and that reads each line of the file and writes it back reversed. The method will not return anything and will not throw any exceptions. For example, if the file **stuff.txt** contained the lines “these are” and “two lines” then after executing *reverseFile*, **stuff.txt** would contain the lines “era eseht” and “senil owt”.

```
public void reverseFile(String fName) {
    String[] contents = new String[100]; // assume 100 lines
    try {
        Scanner in = new Scanner(new File(fName));
        int lineCount = 0;
        while (in.hasNext()) {
            contents[lineCount] = in.nextLine();
            lineCount++;
        }
        in.close();
        PrintWriter out = new PrintWriter(new File(fName));
        String revLine = null;
        for (int n = 0; n < lineCount; n++) {
            revline = "";
            for (int k = 0; k < contents[n].length; k++)
                revline = contents[n].charAt(k) + revline;
            out.println(revline);
        }
        out.close();
    } catch (FileNotFoundException e) {
        System.out.println(e.getMessage());
    }
}
```

9. Add a constructor and a *run()* method to the Java class below named **TestThreads** that extends **Thread**. The class will have one instance variable, an *int* named *sleepTime*. The constructor will have one parameter, a String that names a Thread. When the constructor executes, it will set *sleepTime* to a random integer between 0 and 5000, and then print the name of the thread and how long it will sleep. The *run()* method will make the thread sleep for its *sleepTime* and then print a message when the thread wakes up.

```
public class TestThreads extends Thread {
```

```

private int sleepTime;
public TestThreads(String threadName) {
    sleepTime = (int) (Math.random()*5000.);
    System.out.println(threadName + " will sleep for " +sleepTime);
    this.start();
}
public void run() {
    System.out.println("Thread " +this.getName() + " started");
    try {
        this.sleep(sleepTime);
        System.out.println("Thread " +this.getName() + " woke up");
    }
    catch (InterruptedException e) {
        System.out.println(e.getMessage());
    }
}
public static void main(String[] args) {
    TestThreads t1 = new TestThreads("T1");
}
}

```

10. Write the Javascript for a Web page that has two buttons called “Hot” and “Cold”. When the “Hot” button is clicked, the background of the page will turn Red and a text message in Black will say “Stay Hot!” and when the “Cold” button is pushed, the background of the page will turn Blue and a text message in Black will say “Be Cool!”

```

<html>
  <head>
    <SCRIPT LANGUAGE="JavaScript">
      function hot(code) {
        document.write("Stay Hot!");
        document.bgColor=code;
      }
      function cold(code) {
        document.write("Be Cool!");
        document.bgColor=code;
      }
    </SCRIPT>
  </head>
  <body>
    <form>
      <INPUT NAME="b1" TYPE="button" VALUE="Hot"
        onClick="hot('red')">
      <INPUT NAME="b2" TYPE="button" VALUE="Cold"
        onClick="cold('blue')">
    </form>
  </body>
</html>

```