Question 1. 5 points

1. Write exactly one loop statement - **for** or **while** - that prints this pattern of ten words:

   dog
dog
dog
dog
cat
dog
dog
dog
dog
dog

Answer:
for(int j = 0; j < 10; j++){
    if (j != 4) System.out.println("dog");
    else System.out.println("cat");
}

Question 2. 5 points

Suppose s is some String. Write a loop that prints the characters in s in a row backwards, and also doubled. So if s is "cat", for example, your code should print

   ttaacc

If s is "hee*haw" your code should print

   wwaahh**eeeehh

Answer:
for(int j = s.length()-1; j>= 0; j--){
    System.out.print(s.charAt(j));
    System.out.print(s.charAt(j));
}

Note: System.out.print(s.charAt(j)+s.charAt(j)) doesn't work, but you didn't lose points. However if you had this, and you messed up on the for loop limits, you lost a point.

Question 3. 10 points

Rewrite this code fragment using a while loop.

```java
for(int j = 3; j < 1000; j = 3*j){
    System.out.println(j);
}
```

Answer:
int j = 3;
while (j < 1000){
    System.out.println(j);
    j = 3*j;
}
**Question 4. 10 points**

Suppose a static method in the `Calculate` class called `stringMin` looks like this:
```java
public static int stringMin(String s, String t){
    int ans;
    **your code goes here**
    return ans;
}
```

It is passed two `String` values, `s` and `t`, which you should assume consist only of digits. Thus the method's arguments might be "45697" and "121". `stringMin` should return an `int` value that is the minimum of the numbers that have been represented in `String` form in the method parameters.

Answer:

```java
int m = Integer.parseInt(s);
int n = Integer.parseInt(t);
ans = Math.min(m,n);
```

**Question 5. 25 points**

Write a complete program in a single class called `DigitSearch`. It should read in a `String` value - a word or phrase - call it `s`, and two `int` values - call them `m` and `n` - from the keyboard. If `String` `s` has a digit at any position from `m` through `n`, inclusive, your code should print `true`. On the other hand if there is no digit in that range your code should print `false`. The program should work even if one or both of `m`, `n` fall outside the position bounds of the string.

```java
import java.util.Scanner;

public class DigitSearch{
    public static void main(String[] args){
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter string");
        String s = scan.nextLine();
        System.out.println("Enter two ints");
        int n = scan.nextInt();
        int m = scan.nextInt();
        boolean here = false;
        for(int j = n; j <= m; j++)
            if((j>=0)&&(j<s.length()) && (Character.isDigit(s.charAt(j))))
                here = true;
        System.out.println(here);
    }
}
```

**Question 6. 10 points**

You work for the state of Massachusetts on town voting records and you are preparing a Java class to help with your record keeping. Here is the start of that class.

```java
public class TownVote{
    private String name; // name of town
    private int votes; // votes cast
    private int pop; // town population
```
public TownVote(String name, int v, int pop){
    // constructor body goes here
}

public String getName(){return name;}
public int getVotes(){return votes;}
public int getPop(){return pop;}
}

For this problem your job is to write the body of the class constructor. Write the body of the constructor only; do not alter the constructor header line. In your answer associate name with the name field of the object being created, v with the votes field, and pop with the pop field.

Answer:
this.name = name; voters = v; this.pop = pop;

**Question 7. 10 points**
Assuming that you've written the constructor for this class, now add to the class a public method called fractionVote, which takes no arguments and which returns a double value, the fraction of the town population that has cast votes. Thus if a town has a population of 6000 and 1500 votes are cast, the method should return .25.

Answer:
public double fractionVote(){return ((double)votes/pop);}

**Question 8. 10 points**
Assume you've written the constructor for this class. Now add to the class a public method called highVoteFraction, which is passed one argument, a decimal value, and which returns true or false - a boolean value - if the fraction of the town population that has cast votes exceeds the parameter value.

Answer:
public Boolean highVoteFraction(double cutoff){return(((double)votes/pop) > cutoff);}

**Question 9. 5 points**
Answer:
TownVote townOfAmherst = new TownVote("Amherst", 7654,27134);

**Question 10. 10 points**
Assume again that you've written the constructor for the TownVote class, and suppose TownVote objects t1, t2, and t3 have already been created in VoteDriver. Now add a statement or statements to the driver that print the name of the town among these three with the smallest population. In case of ties, the name of any of the smallest will do.

Answer:
int p1 = t1.getPop(); int p2 = t2.getPop(); int p3 = t3.getPop();
if ((p1 <= p2) && (p1 <= p3)) System.out.println(t1.getName());
else if ((p2 <= p1)&&(p2 <= p3)) SOP(t2.getName());
else SOP(t3.getName());