1.) Write a static method countLetter() that accepts a String str and char c as parameters, and returns the number of times c occurs in str.

```java
static int countLetter(String str, char c){
    int count = 0;
    for(int i = 0; i<str.length(); i++) {
        if(str.charAt(i) == c)
            count++;
    }
    return count;
}
```

2.) What is the output of this program?

```java
public class Question2{
    public static void main(String[] args){
         System.out.println(func1(7));
    }
    public static int func1(int x){
         System.out.println("func1: " + x);
         int y = func2(x % 5);
         return y;
    }
    public static int func2(int a){
         System.out.println("func2: " + a);
         return func3((a * 2));
    }
    public static int func3(int x){
         System.out.println("func3: " + x);
         return (x + 1);
    }
}
```

**Output:**

<table>
<thead>
<tr>
<th></th>
<th>func1: 7</th>
<th>func2: 2</th>
<th>func3: 4</th>
<th>5</th>
</tr>
</thead>
</table>
3.) What is the output of this program?

class Question3{
    public static void main(String[] args){
        Driver a = new Driver("Sarah Lynn");
        Driver b = new Driver("Todd");
        Driver c = new Driver("Diane");

        Driver d = b;
        Driver e = d;

        drive(a);
        drive(b);
        drive(c);
        drive(d);
        drive(e);

        System.out.println(a.toString());
        System.out.println(b.toString());
        System.out.println(c.toString());
        System.out.println(d.toString());
        System.out.println(e.toString());

        System.out.println(a.driverCount);
        System.out.println(e.driverCount);
    }

    public static void drive(Driver z){
        Driver y = z;
        y.miles++;
        z.miles++;
    }
}

class Driver{
    int ID;
    String name;
    int miles;
    static int driverCount = 0;
    Driver(String name){
        this.name = name;
        driverCount++;
        ID = driverCount;
        miles = 0;
    }

    public String toString(){
        return (ID + " " + name + " " + miles);
    }
}
4.) (a) Complete the recursive method recur() below such that the method computes values for the sequence defined by:

\[ \text{recur}(1) = 2 \]
\[ \text{integer } x > 1: \text{recur}(x) = 2 \times \text{recur}(x - 1) \]

```c
int recur(int x) {
    if (x == 1)
        return 2;
    else
        return 2 * recur(x - 1);
}
```

(b) Rewrite this method using iteration.

```c
int recur(int x) {
    int count = 2;
    while (x > 1) {
        count *= 2;
        x--;
    }
    return count;
}
```
5.) Given the Node class shown here, write a method `len()` that takes a Node and returns the length of the linked list, i.e. how many elements are in the list. Complete this method (a) recursively, and (b) iteratively.

```java
class Node {
    Node next;
    int data;
    Node(int d, Node n) {
        data = d;
        next = n;
    }
}
(a)
// recursive
int len(Node n){
    if(n == null)
        return 0;
    else
        return 1 + len(n.next);
}
(b)
// iterative
int len(Node n){
    int count = 0;
    while(n != null){
        count++;
        n = n.next;
    }
    return count;
}
```
6.) The following program should display a single button that when clicked prints “Submitted.” Fill in the blanks to complete the program. Everything you need to fill the blanks is either in the following list of fragments or is a variable in the incomplete program (e.g. window). Some items in the list may be used to fill more than one blank. Every item is used at least once.

(ActionEvent
  ActionListener
  addActionListener
  actionPerformed
  getContentPane
  new Clicked()

import java.awt.event.*;

class Clicked implements ActionListener{
    public void actionPerformed(ActionEvent e) {
        System.out.println("Submitted");
    }
}

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

class SimpleButton {
    public static void main(String[] args) {
        JFrame window = new JFrame("Simple Button");
        Container content = window.getContentPane();
        JButton clicker = new JButton("Submit");
        content.add(clicker);
        clicker.addActionListener(new Clicked());
        frame.pack();
        frame.setVisible(true);
    }
}
7.) Write a method, `fill()`, that takes two parameters, a 2D array of `int` and an `int`. The method returns nothing but fills the array with the value passed as the second parameter. E.g. `fill(myArray, -1);` would fill `myArray` with `-1`s.

```java
void fill(int[][] twoD, int n){
    for(int i = 0; i<twoD.length; i++){
        for(int j = 0; j<twoD[i].length; j++){
            twoD[i][j] = n;
        }
    }
}
```

8.) Write a method `multiply()` that takes two parameters: an array of integers `vector`, and an integer `scale`. The method should not return anything but instead modify `vector`, multiplying each element by `scale`. For example, if `data` was the array `{11, 22, 33}` then after calling `multiply(data,10), data` would contain `{110, 220, 330}`. Only write the method.

```java
void multiply(int[] vector, int scale){
    for(int i = 0; i<vector.length; i++){
        vector[i] *= scale;
    }
}
```
9.) What does this program print?

```java
class Question9 {
    public static void main(String[] args) {
        char[][] data;
        data = {{'0','1','2'},{'3','4'},{'5','6','7','8'}};
        for (int i = data.length-1; i >= 0; i--)
            for (int j = 0; j < data[i].length; j++)
                System.out.print(data[i][j]);
        System.out.println();
    }
}
```

Output:

```
5678
34
012
```
10.) What does the following program print?

```java
import java.util.Scanner;
class Problem10{
    public static void main( String[] args ){
        int a=2, b=3, c=1;
        double x=1.0, y=1.5, z=2.0;
        c = fcn1(a, b);
        y = fcn2(y, a);
        b = fcn3(x, y);
        z = fcn3(c, b);
        System.out.println("a=",a+", b=",b+", c=",c);
        System.out.println("x=",x+", y=",y+", z=",z);
    }

    static int fcn1(int i, int j){
        int k = i+j;
        k *= 2;
        return (k);
    }

    static double fcn2(double t, int n){
        return (t+n);
    }

    static int fcn3(double u, double v){
        return fcn1((int)(u+v), 2);
    }

    static double fcn3(int r, int s){
        return fcn2(r,s);
    }
}
```

Output:
```
a=2, b=12, c=10
x=1.0, y=3.5, z=22.0
```
11.) Write a class `Point` that contains private instance variables `x` and `y`, along with all the implied and necessary methods. The `Point` class has a constructor that initializes `x` and `y`. The `Point` class also has a public method `midPoint()` which accepts a `Point` object `p` as a parameter, and returns a `Point` object that is the midpoint of this and `p`. For points `(x1, y1)` and `(x2, y2)`, the midpoint is `((x1+x2)/2, (y1+y2)/2)`. Finally, `Point` has a method `toString()` that returns a String representation of the `Point` object in the form of `(x, y)`.  

```java
public class Point {
    private int x, y;
    Point(int x, int y){
        this.x = x;
        this.y = y;
    }
    public int getX() {
        return x;
    }
    public int getY() {
        return y;
    }
    public void setX(int x) {
        this.x = x;
    }
    public void setY(int y) {
        this.y = y;
    }
    public Point midPoint(Point p) {
        return new Point((x + p.getX())/2, (y + p.getY())/2);
    }
    public String toString() {
        return "(" + x + ", " + y + ")";
    }
}
```
// The following classes are for the next two questions
class ClassOne {
    int data = 100;
    public int get() {
        return data;
    }
    public int mystery() {
        return get() + 20;
    }
}
class ClassTwo extends ClassOne {
    int data = 200;
    public int get() {
        return data;
    }
}

12.) What does the following program print?

class Question12 {
    public static void main(String[] args) {
        someFunc(new ClassOne());
        someFunc(new ClassTwo());
    }
    static void someFunc(ClassOne one) {
        System.out.println(one.mystery());
        System.out.println(one.get());
    }
}

Output:
120
100
220
200

13.) What does the following program print?

class Question13 {
    public static void main(String[] args) {
        foo(new ClassOne());
        foo(new ClassTwo());
    }
    static void foo(ClassOne one) {
        someFunc(one);
    }
    static void someFunc(ClassOne one) {
        System.out.println("one "+ one.get());
    }
    static void someFunc(ClassTwo two) {
        System.out.println("two "+ two.get());
    }
}

Output:
one 100
one 200