(30) Write a complete program in a single class called StarString, which does the following. First it reads in a line of text from the keyboard. Then it prints to the screen the entered line of text, with the following special condition. Every third character should be replaced with a star
symbol '*'.
For example, if the line is

Abcdefghijklmn   your code should print  *bc*ef*hi*kl*n

And if it’s

The dog barks   your code should print   *he*do* b*rk*

Thus, in any String, the characters at positions 0, 3, 6, and so forth should be replaced with *.

---

```java
import java.util.Scanner; //import Scanner to be used later
//Define class
public class StarString
{
  //set up main
  public static void main(String[] args)
  {
    Scanner scan = new Scanner(System.in); //set up scanner
    String input = scan.nextLine(); //use next line method and assign it to a String
    //for loop to cycle through String and print out every third character
    for(int i = 0; i < input.length(); i++)
    {
      if(i%3 == 0)
      {
        //use print, not print line since you want to retain the format of the string
        System.out.print("*");
      }
      else
      System.out.print(input.charAt(i));
    }
    System.out.println(); //to move the cursor to the next line
  }
```

2. (5) What does this statement print? Put a circle around your answer.
System.out.println((110 % 20) / ((784 % 100) / 10));

110 % 20 = 10  
784 % 100 = 84  
84 / 10 = 8  
10 / 8 = 1 <- Final Answer

3. (5) Write a for loop that prints in a column all of the even numbers from 150 to 66 inclusive except for 132 and 104.

for(int i =150; i<=66; i++)
{
    //Need to use ands since it cannot be either of those numbers and has to be even
    if(i != 132 && i != 104 && i % 2 ==0)
    {
        System.out.println(i);
    }
}

4. (5) What do these statements print? Put a circle around your answer (note: (int)'A' = 65)

    int n = 70;
    if ('A' > n) System.out.println('A'*10); else System.out.println('A'/10);

Since 70 > 65, the else statement kicks off. We get 65 / 10 = 6 <- Final answer

5. (30) Consider the Fence class below:

```java
public class Fence{
    private double length; // in feet
    private int height; // in feet
```
a) The Fence class constructor is missing its body. In the space below, write the body for the Fence class constructor (you don’t need to write out the entire class).

//Attributes on left, input on right
length = len;
height = h;
this.owner = owner; //since both are named the same, have to reference attribute with this

b.) Now write a new class called FenceDriver, which includes main. Inside main make a Fence object called myFence, with owner Dana, length 25.4 feet, and height 3 feet.

//Define class
public class FenceDriver
{
    //set up main
    public static void main(String[] args)
    {
        //create object
        Fence myfence = new Fence(25.4, 3, "Dana");
    }
}

1 Add a method to the Fence class called area, which returns the area of the fence – the product of the fence’s height and length. Just write the method definition – no need to copy over the entire class.

public double area()
A high fence is a fence that is more than 4 feet tall. Add a method to the Fence class called isHighFence, which returns true if the calling object is a high fence (more than 4 feet high), and otherwise returns false.

```java
public boolean isHighFence()
{
    return (height > 4);
}

//also valid
public boolean isHighFence()
{
    if(height > 4)
        return true;
    else
        return false;
}
```

Suppose fenceA and fenceB are two fence objects from the Fence class that have been defined inside main in FenceDriver, and suppose that there has been a mistake: the owner of fenceA should be the owner of fenceB, and the owner of fenceB should be the owner of fenceA. Write several lines of code, also in main, which swap the two owners so that the each fence has its proper owner.

```java
//no need to write main again
//Need a temp variable to hold one name,
```
//otherwise you'll be setting both objects to the same owner
String temp = fenceA.getOwner();
fenceA.setOwner(fenceB.getOwner());
fenceB.setOwner(temp);

6. (5) Write a for loop that prints the phrase “Hello Mother” 100 times in a column. Each line should be numbered, so that your printout should look like:
Hello Mother
Hello Mother
Hello Mother
Hello Mother

....
100 Hello Mother

//Start at 1 for simplicity
for(int i = 1; i<= 100; i++)
{
    System.out.println(i + " Hello Mother");
}

7.(10) Write a for loop that adds up and then prints the sum of the squares of all of the numbers from 100 to 200 inclusive. Thus your code should print the value of this sum: 100^2 + 101^2 + … + 200^2
/Need a variable defined outside the for loop so I can reference it after my for loop
int sumOfSquares = 0;
for(int i = 100; i<= 200; i++)
{
    sumOfSquares+= (i*i); //also valid: sumOfSquares = sumOfSquares + (i*i)
}
System.out.println(sumOfSquares);

8. (5) In the space below carefully write down what this code prints:

String t = “By”;  
for (int j = 5; j < 8; j++){ t = t + t; System.out.println( t);}

//Remember addition just adds the two strings together and we're calling
//println inside the for loop which puts each run on its own row
ByBy
ByByByBy
ByByByByByByByBy

9. (5) Which of the following statements include static method calls. Circle all that apply:
myKid.anotherMonth(); // myKid is an Infant object
System.out.println(Math.pow(2, 5));
if (Character.isLetter(x)) System.out.println(x);
this.age = 14;
String s = scan.nextLine(); // scan is a Scanner object

// A static call is a method that is called on a class not an object
// so pow is called by the Math class we didn't have to create a Math object
// same thing with isLetter

B, C <- Final Answer