Problem 1 15 Pts

Calculate the value taken by variable $x$ resulting from the execution of the code fragment below. Place parentheses to indicate the order of evaluation of the expression, according to the precedence and associativity rules of Java, and indicate all the type conversions, implicit or explicit, that occur during the evaluation.

```java
int m = 100, n = 9;
double x = 10.0;
float z = 1.0F;
x += (float) n / m / 3 + n / -m / 3.0 + z * n / 2.0 - x;
```
**Problem 2**

20 Pts

Write in the space available on this page the code of a method named `norm3D` that takes three `double` numbers (corresponding to the coordinates of a 3D vector) as parameters and returns the norm of the vector, that is, the square root of the sum of the squares of these numbers. The appendix give the part of the Java API specifications that are relevant to this problem.

**Problem 3**

10 pts

Identify as many syntax error as you can in the following code

```java
public static void myMethod(int m, float val)
{
    int m, n;
    float x = new float(3.0)
    x += 0.5 * val;
    n++;
    x + val = n * * 2;
    return x;
}
```
Problem 4  
Consider the following (really ugly, but valid) class implementation

```java
public class SillyClass {
    int m, n;

    public SillyClass() {
        m = 0;
        n = 1;
    }

    public int firstMethod(int n) {
        m -= 1;
        n *= 5;
        return m + n;
    }

    public int secondMethod(int m, int k) {
        m += 2 * this.m;
        this.n += n * k;
        return m;
    }
}
```

What would be the value of variables `l` and `m` in the `main` method and of instance variables `m` and `n` of object `myObj` after the execution of the following code:

```java
public static main void(String []arg) {
    SillyClass myObj = new SillyClass();
    int l = 2, m = 10;
    m = myObj.firstMethod(m);
    l = myObj.firstMethod(m);
    m = myObj.secondMethod(l, m);
}
```
Problem 5 30 Pts
You are asked to implement a class Vector3D to store and manipulate information relative to 3D vectors. A 3D vector is represented by three coordinates (double numbers). You do not need to implement access methods for this class (all instance variable will be public and therefore directly accessible), but the following should be implemented:

Constructors:
- a default constructor that initializes the three coordinates to zero
- a constructor that receives as parameter three double values with which to initialize the three coordinates of the vector.

Methods:
- a method named `scale` that multiplies the vector’s three coordinates by the double number received as parameter.
- a method named `normalize` that divides all coordinates of the vector by the vector’s norm (if it is not equal to zero). You can invoke for this purpose the method of Problem 3 (even if you haven’t answered the question).
- a method named `dotProduct` that receives as parameter another Vector3D object and returns the dot product of the two vectors, that is, the sum of the products of their corresponding coordinates. For example \((10, 2, 3) \cdot (4, 1, 5) = 10 \cdot 4 + 2 \cdot 1 + 3 \cdot 5 = 57\).
- a method name `addVector` that receives as parameter a Vector3D object and returns a new Vector3D object that is the sum (coordinate by coordinate) of this vector with the one received.