**Java OOP(Object Oriented Programming) Concepts**

**Object-Oriented Programming** or **Java OOPs** concept refers to programming languages that use objects in programming. They use objects as a primary source to implement what is to happen in the code. Objects are seen by the viewer or user, performing tasks you assign.

**Object-oriented programming** aims to implement real-world entities like **inheritance**, **hiding**, **polymorphism**, etc. in programming. The main aim of OOPs is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

Numbers.java

// Source code is decompiled from a .class file using FernFlower decompiler.

class Numbers {

   private int a;

   private int b;

   Numbers() {

   }

   public void sum() {

      System.out.println(this.a + this.b);

   }

   public void sub() {

      System.out.println(this.a - this.b);

   }

   public static void main(String[] var0) {

      Numbers var1 = new Numbers();

      var1.a = 1;

      var1.b = 2;

      var1.sum();

      var1.sub();

   }

}

Numbers.class

// Source code is decompiled from a .class file using FernFlower decompiler.

class Numbers {

   private int a;

   private int b;

   Numbers() {

   }

   public void sum() {

      System.out.println(this.a + this.b);

   }

   public void sub() {

      System.out.println(this.a - this.b);

   }

   public static void main(String[] var0) {

      Numbers var1 = new Numbers();

      var1.a = 1;

      var1.b = 2;

      var1.sum();

      var1.sub();

   }

}

It is a simple example showing a class Numbers containing two variables which can be accessed and updated only by instance of the object created.

