Lab 7:

```
class Generic <T, U> {
    T ob1;
    U ob2;
    Generic (T x, U y) {
        ob1 = x;
        ob2 = y;
    }
    T get ob1 () {
        return ob1;
    }
    U get ob2 () {
        return ob2;
    }
    void display () {
        System.out.println ("Ob1:" + get ob1());
        System.out.println ("Ob2:" + get ob2());
    }
    U join () {
        if (ob1 instanceof Integer && ob2 instanceof Integer)
        {
            int i1 = (Integer) get ob1();
            int i2 = (Integer) get ob2();
            return (U) new Integer (i1 + i2);
        }
        else if (ob1 instanceof Double && ob2 instanceof Double)
        {
            double d1 = (Double) get ob1();
            double d2 = (Double) get ob2();
            return (U) new Double (d1 + d2);
        }
        else if (ob1 instanceof String && ob2 instanceof String)
        {
            String s1 = (String) get ob1();
            String s2 = (String) get ob2();
            return (U) new String (s1 + s2);
        }
```

```java
        else {
            return (U) new String("ERROR: ob1 & ob2 Type Mismatch...");
        }
    }
}

public class lab7 {
    public static void main(String [] args) {
        Generics < Integer, Integer > iobj = new Generics < Integer, Integer >
                                                                    (5, 4);
        iobj.display();
        System.out.println(" Sum:" + iobj.join());
        System.out.println();

        Generics < Double, Double > dObj = new Generics < Double, Double >
                                                            (3.05, 4.02);
        dObj.display();
        System.out.println(" Sum:" + d.obj.join());
        System.out.println();

        Generics < String, String > sObj = new Generics < String, String >
                                                ("Hello.", "How are you?");

        sObj.display();
        System.out.println("Concatanation:" + sObj.join());
        System.out.println();
    }
}
```