1. Consider the following code fragments. For each, if there is a compiler error, identify where it occurs.

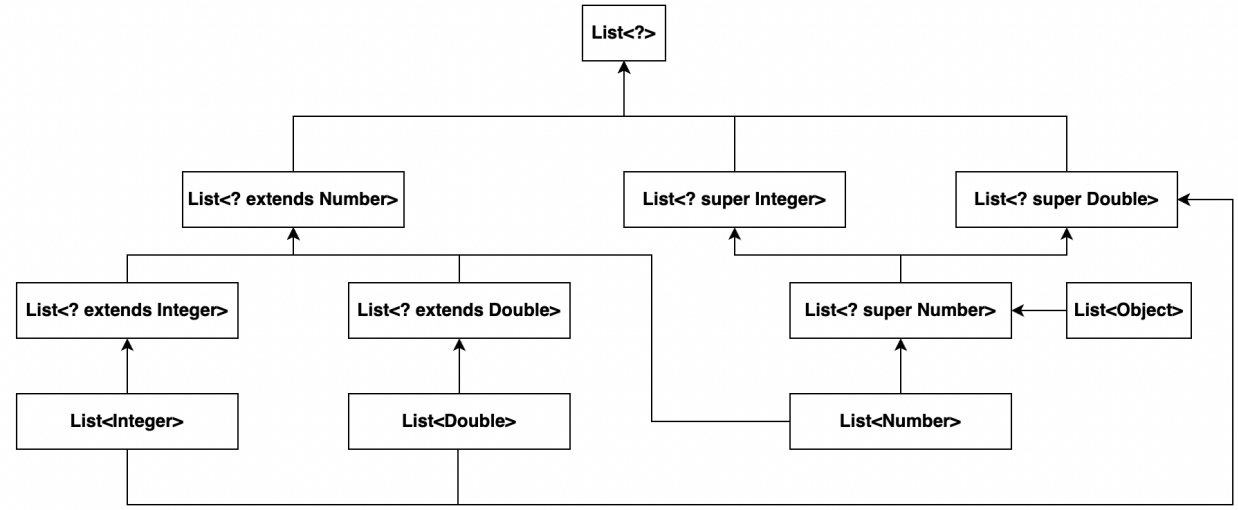
a. First fragment:

```
List<Integer> ints = new ArrayList<>();
ints.add(1);
ints.add(2);
List<Number> nums = ints;  --> compiler error
nums.add(3.14);
```

b. Second fragment:

```
List<Integer> ints = new ArrayList<>();
ints.add(1);
ints.add(2);
List<? extends Number> nums = ints;
nums.add(3.14);  --> compiler error
```

3. Draw a class diagram showing the inheritance relationships among the following types: List<Integer>, List<Number>, List<? extends Integer>, List<? extends Number>, List<? super Integer>, List<? super Number>, List<?>, List<Object>, List<Double>, List<? extend Double>, List<? super Double>



4. Recall the definition of sum given in the slides:

```
public static double sum(Collection<? extends Number> nums) {
    double s = 0.0;
    for(Number num : nums) s += num.doubleValue();
    return s;
  }
```

a. Is there a compiler error in the following lines of code? If so, where?

```
List<Integer> ints = new ArrayList<>();
ints.add(1);
ints.add(2);
List<? extends Number> nums = ints;
double dbl = sum(nums);
nums.add(3.14); --> compiler error
```

b. Is there a compiler error in the following lines of code? If so, where?

```
List<Object> objs = new ArrayList<>();
objs.add(1);
objs.add("two");
List<? super Integer> ints = objs;
ints.add(3);
double dbl = sum(ints); --> compiler error
```