1). Imagine two classes, A and B.
Class A has a private data field theData and the following methods:
public void w();
public void x();
protected void y();
private void z();
Class B extends class A and has the following methods:
public void x();
protected void r();
private void s();
Suppose that the client declares instances of these classes as follows:
A inA = new A();
B inB = new B();
a. Which of the objects inA and inB can the client use to directly access the field the-Data?
b. Which of these objects can the client use to invoke the method w?
c. Which of these objects can the client use to invoke the method y?
d. Which of these objects can the client use to invoke the method r?
e. Which of these objects can the client use to invoke the method z?
f. Which method definitions can invoke the method y?
g. Which method definitions can invoke the method r?
h. Which method definitions can invoke the method z?
i. Which version of the method x does inB.x() invoke?
j. Which methods are available to the implementation of the class B?
k. Which methods are available to clients of B?

2). Consider the following Java statements:
public static Student s1 = new GraduateStudent(...);
public static void main(String args[])
{
    Student ug = new UndergradutateStudent(...);
    GraduateStudent grad = new GraduateStudent(...);
    winner(ug);
}
public static void winner(Student pupil)
{
    System.out.println("And the winner is ");
pupil.displayAt(1);
}
a. What are the static and dynamic types for each variable when the code is executed?
b. If we change the call winner(ug) to winner(s1), what are the resulting static and
dynamic types for pupil?
3). If myList is an empty list of strings, what does it contain after the following
statements execute? Please show the list content after each statement.

```java
myList.add("alpha");
myList.add(1, "beta");
myList.add("gamma");
myList.add(2, "delta");
myList.add(4, "alpha");
myList.remove(2);
myList.remove(2);
myList.replace(3, "delta");
```

4). Suppose that you have a list that is created by the following statement:

```java
ListInterface<Double> quizScores = new AList<Double>();
```

Imagine that someone has added to this list the quiz scores received by a student
throughout a course. The professor would like to know the average of these quiz scores,
ignoring the lowest score.

**a.** Write Java statements at the client level that will find and remove the lowest score
in the list.

**Answer:**

```java
int positionOfSmallest;
double smallest, toCompare;
int count = quizScores.getLength();
if (count > 0) {
    positionOfSmallest = 1;
    smallest = quizScores.getEntry(1);
    for (int i = 2; i <= count; i++) {
        toCompare = quizScores.getEntry(i);
        if (toCompare < smallest) {
            smallest = toCompare;
            positionOfSmallest = i;
        }
    }
    quizScores.removeEntry(positionOfSmallest);
}
```

**b.** Write Java statements at the client level (similar to above example in a) that will
compute the average of the scores remaining in the list.