Test 2

For each question, please **submit** **ALL OF YOUR SOURCE CODES** **and** the **SCREENSHOT** of the executed program.

Question 1:

Create package with your initial test21. (e.g. James Bond use "**jbtest21**").
The completed program should run as shown below:

TestMain21.java > TestMain21

```java
import java.util.Random;

class TestMain21 {
    Run | Debug
    public static void main(String[] args) {
        int[] data = { 2, 3, 7, 8, 10, 12, 30, 32, 33, 33, 35, 35, 36, 48, 50, 78, 89, 91, 98, 99};
        BSearch s = new BSearch(data);
        Random rand = new Random();
        for (int i = 0; i < 20; i++) {
            int rno = rand.nextInt(100);
            int index = s.search(rno);
            if (index == -1) {
                System.out.printf("%d cannot be found!\n", rno);
            } else {
                System.out.printf("%d is found at index %d\n", rno, index);
            }
        } // end for
    } // end main()
} // end TestMain21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
66 cannot be found!
21 cannot be found!
52 cannot be found!
66 cannot be found!
91 is found at index 17
13 cannot be found!
83 cannot be found!
syang2@mbp14 test21 %
```

Complete **BSearch.java** codes to run the program as shown in the above screenshot. Note the comparing number will be generated as a random. The output number may not have the exact same numbers as shown in the above screenshot.

*** public int search(int no, int start, int end) method should be complete with recursive method ***

Question 2:

Create package with your initial test22. (e.g. James Bond use "**jbtest22**").
The completed program should run as shown below:

TestMain22.java > TestMain22 > main(String[])

```java
public class TestMain22 {
    Run | Debug
    public static void main(String[] args) {
        List<String> currentNode;
        List<String> head = new List<>("James");

        currentNode  = head;
        List<String> tempNode = new List<>("Jill");
        currentNode.addNode(tempNode);

        currentNode = currentNode.getNextNode();
        tempNode = new List<>("Tilly");
        currentNode.addNode(tempNode);

        currentNode = currentNode.getNextNode();
        tempNode = new List<>("Felix");
        currentNode.addNode(tempNode);

        // print list content
        currentNode = head;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
syang2@mbp14 test22 %
syang2@mbp14 test22 %  cd /Users/syang2/Documents/java_projects/test/src/test22 ; /usr/bin/en
ts/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/syang2/Library/Applicatio
5c7bbbd9e370e/redhat.java/jdt_ws/test22_20fd96d0/bin TestMain22
James Jill Tilly Felix
=====
10.1 30.3 50.5 70.7 90.9
syang2@mbp14 test22 %
```

Complete **List.java** codes to run the program as shown in the above screenshot.
*** The List class should use Generic type T and should have the class constructor, getValue(), hasNext(), addNode(), and getNextNode() methods. ***

Question 3:

Create package with your initial test23. (e.g. James Bond use "**jbtest23**").
The completed program should run as shown below:

TestMain23.java > ...

```java
public class TestMain23 {
    Run | Debug
    public static void main(String[] args) {
        PersonStack ps = new PersonStack();
        ps.push(new Person("James"));
        ps.push(new Person("Jill"));
        ps.push(new Person("Felix"));
        ps.push(new Person("Simmons"));
        System.out.println("===== output =====");
        while (!ps.isEmpty()) {
            System.out.println(ps.pop().getName() + " ");
        }
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ts/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/syang2/Library/
43db22eaf309c/redhat.java/jdt_ws/test23_20fd96d1/bin TestMain23
===== output =====
Simmons
Felix
Jill
James
syang2@mbp14 test23 %
```

Complete **PersonStack.java** codes to run the program as shown in the above screenshot.
*** The PersonStack class should the given Person class and should have the class constructor, isEmpty (), push(), and pop() methods. ***

Question 4:

Create package with your initial test24. (e.g. James Bond use "**jbtest24**").
The completed program should run as shown below:

TestMain24.java > TestMain24

```java
public class TestMain24 {
    Run | Debug
    public static void main(String[] args) {
        Person[] data = new Person[5];
        data[0] = new Person("James");
        data[1] = new Person("Jill");
        data[2] = new Person("Tilly");
        data[3] = new Person("Felix");
        data[4] = new Person("Simmons");

        InsertionSort s = new InsertionSort(data);
        s.sort();

        System.out.println("=== output ===");
        for (int i = 0; i < data.length; i++) {
            System.out.print(data[i].getName() + " ");
        }
        System.out.println("\n=============");
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
syang2@mbp14 test24 %
syang2@mbp14 test24 %  cd /Users/syang2/Documents/java_projects/test/src/test24 ; /usr/bin/env /
ts/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/syang2/Library/Application\
d7ab778200e73/redhat.java/jdt_ws/test24_20fd96d2/bin TestMain24
=== output ===
Felix James Jill Simmons Tilly
=============
syang2@mbp14 test24 %
```

Complete **InsertionSort.java** codes to run the program as shown in the above screenshot.
*** The InsertionSort class should use the given Person class and should have the class constructor and sort() methods. ***
*** hint ***
When comparing the order of two Person class objects, use the isGreaterThan() method in the Person class to check. The isGreaterThan() method will return true, if the person object is greater than the person object in the parameter.
For example,

```
Person p1 = new Person("A");
Person p2 = new Person("B");
p1.isGreaterThan(p2) will return false
p2.isGreaterThan(p1) will return true
```