

1. True/False [15 pts]

Answer the following questions True or False. Be sure to write out True and False, NOT just a T or F.

- 3 (a) Abstract classes can be instantiated under certain conditions.
- 3 (b) `java.lang.Object` is the universal parent (superclass) of all classes
- 3 (c) Consider the following code:

```
public class Test3{
    public void method1(){

        public void method1(int i){}

        public void method1(int i, int j){}
    }

```

method1 is overloaded in class Test3
- 3 (d) You can extend more than one class, for example:

```
public class Foo extends Bar, Goo {

```

is legal.
- 3 (e) You can implement more than one interface, for example:

```
public class Foo implements Bar, Goo {

```

is legal.

2. Polymorphism/Dynamic Binding[20 pts]

```
public class Car {
    public void revEngine() {
        System.out.println("Zoom, zoom");
    }
}
public class BMW extends Car {
    private String seriesNum;

    public BMW(String seriesNum) {
        this.seriesNum = seriesNum;
    }
    public void printSeries() {
        System.out.println(seriesNum);
    }
}
public class Mercedes extends Car {
    public void revEngine() {
        System.out.println("Zoom, zoom");
    }
}
public class Toyota extends Car {
}
```

Example:

```
Car [] garage = new BMW("6 series"), new Mercedes(), new Toyota();
```

Write a method called **testCars**. This method will take in a Car array like the example array given above. The method will call revEngine on all cars in the array. If a Car is a BMW, also call the printSeries method. Your code should not cause any exceptions.

```
public void testCars(Car[] garage) {
```

3. **LinkedList Coding [25 pts]**

- 15 (a) Write a class `LinkedListNode`. The class must contain data and a link to the next node. Make sure that instance variables (`data`, `next`) have accessors and modifiers. Write a constructor which takes a `Comparable Object` as a parameter and correctly initializes the instance variables `data` and `next`.
- ```
public class LinkedListNode{
```

- 10 (b) Does the **addToLast** method below work. If not FIX it.

```
public class LinkedList{
 LinkedListNode head;

 public void addToLast(Object element){
 LinkedListNode temp = new LinkedListNode(element);

 if(head == null)

 head = temp;

 else {

 while(head.getNext() != null) {

 head = head.getNext();

 }

 head.setNext(temp);

 }
 }
}
```

#### 4. Exceptions [ 20 pts ]

- 10 (a) Write your own exception class called `TooBigException` which extends the class `Exception`.
- 10 (b) Write a method called `isToBig` which takes in an `int`. If the input value is greater than ten throw a `TooBigException`. If the value is less than or equal to ten, print out "Your number is okay". Remember that methods which throw checked exceptions must have a `throws` clause.

## 5. File IO [ 20 pts ]

A Misguided C programmer wrote the following code. With the given input file, he expected the following output, but received something different.

| Input File | Expected Output | Actual Output |
|------------|-----------------|---------------|
| I am a     | I am a          | Ramblin       |
| Ramblin    | Ramblin         | from          |
| Wreck      | Wreck           | Tech          |
| from       | from            |               |
| Georgia    | Georgia         |               |
| Tech       | Tech            |               |

Fix the following code so that the output the programmer receives is the expected output. You may make changes on the code itself.

API Excerpt:

```
class FileReader
 public FileReader(String filename) //opens the file filename
 public int read() // reads the next unicode char from file (eof=-1)
 public void close() //closes the file
class BufferedReader
 public BufferedReader(Reader in)//create buffered character stream
 public String readLine()//read a line of input (eof=null)
 public void close()//closes the file

import java.io.*;
public class MisguidedIOA{

 public MisguidedIOA(){
 try{
 BufferedReader br = new BufferedReader(new FileReader("input.txt"));

 while(br.readLine() != null) {

 System.out.println(br.readLine());

 }

 br.close();

 } catch(Exception e){System.out.println("Wow");}
 }

 public static void main(String[] args){
 MisguidedIOA myIO = new MisguidedIOA();
 } }
```

**6. Extra Credit – Sorting [10 pts ]**

Write out the intermediate results for each step of sorting the following data:

5 3 2 8 1

3 (a) for selection sort:

3 (b) for insertion sort:

4 (c) for mergesort: