STUDENT ID#____________________________

CS 207, Spring 2016, Programming II Exam II
Thursday, March 17, 2016

Instructions:

1. **Do not turn this page until told to do so.**

2. This exam is ***closed book*** and ***closed notes***.

3. ***Write your STUDENT ID Number on every page***. If you do not write your ID Number on a certain page, you ***will not receive credit for the question on that page!***

4. ***Do not write your name on any page (except this one)***. If you write your name on a certain page, ***you will not receive credit for the question on that page!***

5. There are **4** questions on the exam. Once you start, verify you have all 4 questions.

6. You must give your answer to a question on the ***same page that the question appears*** **or** ***the page where you are asked to write your answer.*** If you put your answer on a different page, ***you will not receive any credit for it!***

7. For problems that ask you to write a ***method***, you must write the method header ***exactly*** as shown, but you do not need to write `main()`.

8. For problems that ask you to write a ***program***, you should write the `main()` method ***only***—you can assume the following import statement and keyboard declarations.

```
          import java.util.*;

          Scanner keyboard = new Scanner (System.in);
     or   Scanner kb = new Scanner (System.in);
     or   Scanner input = new Scanner (System.in);
```

9. For tracing problems, assume that any appropriate import statements are provided.

10. You may use "SOP" as an abbreviation for "`System.out.print`" and "SOPln" or "SOPL" for "`System.out.println`".

11. You do not need to do any error checking of input values, ***unless the problem specifically asks you to do so!***

12. If you are caught looking at other papers or communicating with other students in any way, you will receive an **F** for the course.

STUDENT ID#____________________________

Question 1 (20 pts)

Write a method named `removeElement`, which takes a 2D `int` array named `a` and two integers `r` and `c` as parameters and returns a new 2D `int` array with the element at row `r` and column `c` of `a` removed.

1. You may **not** modify the original array `a`.
2. The array being returned may not have `null` values.
3. Your code should be able to handle ragged arrays.
4. If there is no element at row `r` and column `c`, return the original array `a`.
5. You can assume that each row in `a` will have at least two elements.
6. Sample usage is provided below.
7. Complete and place your code in the box on the next page. **Any code on this page will not be considered as part of your solution.**

```
int[][] b = { { 1, 4, 5 },
              { 11, -3 } };

removeElement(b, 1, 0) returns:
{ { 1, 4, 5},
  { -3 } }
```

```
int[][] c = { { -5, -7 },
              { 2, 13 } };


removeElement(c, 4, 2) returns:
{ { -5, -7 },
  { 2, 13 } };
```

```
int[][] d = { { 1, 4, 5, 8 },
              { 11, -3, -20, 3 },
              { 7, 7, 2, 0 } };

removeElement(d, 2, 2) returns:
{ { 1, 4, 5, 8 },
  { 11, -3, -20, 3 }
  { 7, 7, 0 } }
```

```
int[][] e = { { -1, 11, 3 },
              { 1, 2, 3 },
              { 4, 4, 4 } };

removeElement(e, 1, 3) returns:
{ { -1, 11, 3 },
  { 1, 2, 3 },
  { 4, 4, 4 } };
```

STUDENT ID#____________________________

Question 1 (continued)

```
public static int[][] removeElement(int[][] a, int r, int c)
{











}
```

STUDENT ID#____________________________

Question 2 (20 pts)

Write a method, `findLa`, which takes a `String s` as a parameter, and does not return anything. The method should do the following:

1. Print out the words in `s` that have the exact substring `"la"` followed by the index at which `"la"` first occurs in that word.
2. If no words have the substring `"la"`, print out the word followed by a hyphen `"-"`.
3. You may not use more than one for-loop and one conditional (if) statement.
4. Sample usage is provided below. Your output must match the sample output below.

```
String s1 = "Do not label";

findLa(s1) prints:
Do - not - label 0
```

```
String s2 = "Obladi Oblada life goes on";

findLa(s2) prints:
Obladi 2 Oblada 2 life - goes - on -
```

```
String s3 = "Last is awesome";

findLa(s3) prints:
Last - is - awesome -
```

```
String s4 = "availability flames jetlag";

findLa(s4) prints:
availability 4 flames 1 jetlag 3
```

```
public static void findLa(String s)
{

}
```

STUDENT ID#____________________________

Question 3 (20 pts)

Based on the `Person` class provided below, design a properly encapsulated subclass named `Employee` that inherits from `Person` and has the following:

1. A `String` instance variable named `office` (e.g. `"LWH 2222"`) and a `String` instance variable named `hourlyWage` (e.g. `"$45.15"`).
2. A constructor that takes three `Strings` and a `double` as parameters for `name`, `email`, `office`, and `salary` and sets the instance variables of both the parent class and the Employee class.
3. An overridden `toString()` method that returns the concatenation of the `name` and `salary` instance variables.
4. A method named `paycheck` that takes one parameter, an `int` named `hours`, and returns a `double`. The method should calculate and return the wage (`hours x hourlyWage`). You can assume that the `hourlyWage` instance variable will always have a format of a dollar sign followed by several digits.
5. Do **not** add additional methods or variables for either class. Figure out how to do everything based on what has been provided.
6. Complete and place your code in the first box on the next page. **Any code on this page will not be considered as part of your solution.**

Complete the `EmployeeTest` class (with the main method) in the second box on the next page. The class should do the following:

1. Create an `Employee` object and pass in `"Fred"` for `name`, `"RR 205"` for `office`, and `"$38.50"` for `salary`. You may use a reference variable name of your choice.
2. Call the `toString` method and print the result.
3. Call the `paycheck` method, pass in a value of `20` for the parameter and print the result.

```
public class Person
{
    private String name;

    public Person(String n)
    {
        this.name = n;
    }

    @Override
    public String toString()
    {
        return this.name;
    }
}
```

}

STUDENT ID#____________________________

Question 3 (continued)

```
public class EmployeeTest
{
    public static void main(String[] args)
    {











    }
}
```

STUDENT ID#___________________________

Question 4 (20 pts) What is the **exact** output of the `main` method in the `NotesTest` class?

```java
public class Note
{
   public Note()
   {
      System.out.println("Note");
   }

   public void ding()
   {
      System.out.println("Ding1");
   }

   public void honk()
   {
      System.out.println("Honk1");
   }
}
```

```java
public class QuarterNote extends Note
{
   public static String n = "";

   public QuarterNote()
   {
      super();
      n = n + "QN";
      System.out.println(n + "  "
                           + "Quarter");
   }

   public void bell()
   {
      super.ding();
      n = n + "D";
      System.out.println(n);
      System.out.println("Bell1");
   }
}
```

```java
public class HalfNote extends QuarterNote
{
  public void honk()
  {
    super.ding();
    System.out.println("Honk2");
    n = n + "HN";
  }
}
```

```java
public class NotesTest
{
  public static void main(String[] args)
  {
    Note n1 = new HalfNote();
    Note n2 = new QuarterNote();
    QuarterNote n3 = new HalfNote();

    n1.ding();
    n2.honk();
    n3.bell();

    ((QuarterNote) n2).bell();
  }
}
```