CS 307 – Midterm 1 – Summer 2002

Name____________________________________
Last 4 digits of SSN / Student ID ______________
Class Unique ID ___________________________

Instructions:
1. There are 4 questions on this test.
2. You will have 3 hours to complete the test.
3. You may not use a calculator.
4. When code is required, write Java code.
5. Please make your answers legible.
6. The style guide is not in effect, except as noted.
7. Unless noted on the question, you may not using any of the classes or methods from the Java
standard library.

1. (2 points each, 30 points total) Java Mechanics. For all parts, what is the output of the code
fragment? Write your answer on the line provided. Consider each piece of code in isolation. If an
error would occur answer "syntax error" or "runtime error" depending on what type of error it
would be.

A. _________________________________________________

```java
int limit = 13;
int[] intList = new int[limit / 2];

for (int i = 0; i < intList.length; i++)
{  if (i % 3 == 0)
   intList[i] = i;
   else
   intList[i] = -1;
}

for (int i = 0; i < intList.length; i++)
   System.out.print( intList[i] + " ");
```

B. 
```
int x = 17;
int y = 3;

double a = x / y;
System.out.println(a);
```

C. 
```
int i = 0;
int j = 13;

int k = j / i;
System.out.println("k = " + k);
```

D. 
```
int i = 19;
int j = 5;

int k = i % j;
System.out.println(k);
```

E. 
```
//Constructor for x, y, width, height
Rectangle r1 = new Rectangle(5, 10, 15, 20);
Rectangle r2 = new Rectangle(5, 10, 15, 20);

System.out.println(r1 == r2);
```

F. 
```
// two Rectangle objects are "equal" if their
// x, y, width, and height values are all equal

//Constructor for x, y, width, height
Rectangle r1 = new Rectangle(5, 10, 15, 20);
Rectangle r2 = new Rectangle(5, 10, 15, 20);

System.out.println(r1.equals(r2));
```
G. // two Rectangle objects are "equal" if their x, y, width, and height are all the same

// Constructor for x, y, width, height
Rectangle r1 = new Rectangle(5, 10, 15, 20);
Rectangle r2 = new Rectangle(5, 10, 15, 20);
Rectangle r3 = r1;

// set r3's width and height to new values
r3.setSize(100, 200);

System.out.println( r1.equals(r2) );

H. The following two methods appear in the same class. Assume method two has just been called. What is the output?

```java
public void one(int x, int y)
{
    x = y;
    y = 10;
}

public void two()
{
    int a = 3;
    int b = 5;
    System.out.print( a + " " + b);
    one(a, b);
    System.out.print( " " + a + " " + b );
}
```
I. 

```java
String[] sList = new String[4];
sList[0] = "Cardinals";
sList[1] = "Braves";
sList[2] = "Mariners";

// the length method for Strings returns the number of
// characters in the String.

int total = 0;

for(int i = 0; i < sList.length; i++)
    total += sList[i].length();

System.out.println( total );
```

J. 

```java
int[][] mat = new int[5][4];

System.out.println( mat[3].length );
```

K. 

```java
double a = 3 * 5 + 2 * 15 / 3 - 1;
System.out.println( a );
```

L. 

```java
int[] list = { 2, 4, 6, 8, 10, 12, 14, 16 };
int x = list[ list.length - 3 ] * list[ 7 / 2 ] + list[ list[0] ];
System.out.println( x );
```
M. int rows = 4;
   int columns = 5;
   int[][] mat = new int[rows][columns];
   for(int r = 0; r < rows, r++)
       for(int c = 0; c < columns; c++)
           mat[r][c] = r + c;
   System.out.println(mat[0][2] + " " + mat[rows - 2][columns - 1]);

N. int[] list = {2, 3, 5, 7};
   System.out.println( list.length );

O. final int limit = 6;
   int total = 0;
   for(int i = 0; i < limit; i++)
   {  total += i;
      limit--;
   }
   System.out.println( total );
2. Matrices: (25 points) Consider a two dimensional array of integers that represent the temperatures at various grid locations. Write a method that determines the largest temperature differential between a specified cell and all cells within a specified range.

Consider the following matrix:

```
 10  20  30  25  25  30
 30  25  35  30  35  35
 30  20  30  35  30  25
 35  25  35  40  35  30
```

Assume the specified cell is (1, 2). The value of this cell is the largest font in the above diagram and is bolded and underlined. Assume the range is 2. All of the cells in italics (also in a large font) must be checked to mind the maximum temperature difference, above or below, the cell (1, 2). Note, the cell and the range may lead to a cell that does not actually exist. The result of this method call would be 25, the difference between cell (0, 0) and cell (1, 2) is the greatest for a range of 2 and cell (1, 2). You may use the Math.abs method:

```
static int abs(int a)
    Returns the absolute value of an int value.
```

The method is called via `Math.abs(intVariableOrExpression);`

Complete the following method

```java
/*    pre: temps is a non-null, rectangular matrix. (all rows have the same number of columns).
     temp.length > 0 and temp[0].length > 0
     0 <= row < temp.length, 0 <= col < temp[0].length
     range > 0

     post: return the largest temperature difference between the cell specified by row and col and within range cells.
*/
public int getMaxDiff(int[][] temps, int row, int col, int range)
```

// complete the method on the following page
/*  pre: temps is a non-null, rectangular matrix. (all rows have
the same number of columns).
temp.length > 0 and temp[0].length > 0
0 <= row < temp.length, 0 <= col < temp[0].length
range > 0

post: return the largest temperature difference between the
cell specified by row and col and within range cells.
*/
public int getMaxDiff(int[][] temps, int row, int col, int range)
3. Arrays (20 points) Write a method to rotate or shift an array by a specified number of spots.

Here is the original array:

| 2 | 3 | 5 | 7 | 11 | 13 | 17 | 19 | 23 | 29 | 31 | 37 |

If it is to be shifted by 5 the result would be:

| 19 | 23 | 29 | 31 | 37 | 2 | 3 | 5 | 7 | 11 | 13 | 17 |

Notice all elements are shifted "forward" 5 spots. Elements at the end are wrapped around to the front:

Complete the following method:

```java
/* pre: list != null, n > 0
   post: returns an array shifted n spots */
public int[] shiftArray(int[] list, int n)
```
This page intentionally blank. You may use it for scratch paper.
4. Using objects (25 points). Write a method that returns an array of the same Strings, except with all vowels removed. You may use the length and charAt method from the String class as well as the concatenation operator. You may also create new Strings using String literals such as

```java
String s = "name" + someOtherStringVar;
```

You may not use any other methods from the String class.

Vowels are the characters 'A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o', and 'u'.

```java
/*
   pre: org != null, for all n such that 0 <= n < org.length,
   org[n] != null
   post: return an array of Strings equal to org except all
   vowels removed. org is not altered.
*/
public String[] noVowels(String[] org) {
    return new String[org.length];
```

<table>
<thead>
<tr>
<th>char</th>
<th>charAt(int index)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns the character at the specified index.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th>length()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns the length of this string.</td>
</tr>
</tbody>
</table>
This page intentionally blank. You may use it for scratch paper.