Final
CS 1063 - Fall 2014
December 13, 2014
100 points total

Your Name _____

Your Instructor and Section _____

I. (10 points, 1 point each) Match each of the terms on the left by choosing the upper case letter of the best answer on the right and putting it next to the lower case letter in the space provided.

___ a. array	A. a value passed to a method
___ b. assignment statement	B. holds multiple values of the same type
___ c. declaration	C. the part of a program where a declaration is valid
___ d. fencepost loop	D. group of statements with a name
___ e. index	E. a single command that can be executed
___ f. method	F. the first or last value requires special processing
___ g. parameter	G. specifies a variable with a name and type
___ h. reference	H. the location of an array or object
___ i. scope	I. a location in an array or String
___ j. statement	J. stores a value in a variable

II. (10 points) The following program has several errors. For each error, circle where the error occurs, label the circle with a letter: A, B, C, etc. Then for each letter, indicate how you would fix the error.

```
import java.util.*;
public class BuggyProgram {
  public static void main(String args) {
    console = Scanner(System.in);
    x = 5;
    int y;
    System.out.println("Enter an integer:");
    y = nextInt();
    int z = addThem(int x, int y);
    System.out.println("Adding the numbers: " + x + y);
    System.out.println("The sum is z");
    x = addThem(x,z);
    System.out.println("This sum is "+x);
  }
  public static void addThem(x, y) {
    return x + y;
  }
}
```

III. (10 points) What is printed by the following program? Mark your answer clearly.

```
public class StringMystery {
  public static void main (String[] args) {
    String s = "ThisIsATest";
    String t = "IsA";
    int x = 3;
    int y = 5;
    y++;
    System.out.print(s.substring(x,y));
    System.out.print(s.indexOf(t));
    System.out.print(s.indexOf(t.toLowerCase()));
    System.out.print(s.charAt(y));
    System.out.print(s.length());
    System.out.println("Done:" + x + y);
  }
}
```

IV. (10 points) Write a method that takes a single string as a parameter. The method prints the parameter on one line and then creates a new string which is identical to the parameter, but with all occurrences of the character `'a'` replaced with `'x'`. It prints the new string on a separate line and returns the new string.

V. (10 points) What is printed by the following program? Mark your answer clearly.

```
public class MysteryLoops {
  public static void main(String[] args) {
    int x = 5;
    int y = 12;
    if (x + 2 < y - 5)
      System.out.println("This is case 1");
    else
      System.out.println("This is case 2");
    while (x < y) {
      System.out.println(x + " and " + y);
      x = x + 3;
      y = y + 1;
    }
    System.out.println("After loop: " + x + " and " + y);
  }
}
```

VI. (10 points) Write a complete program that prompts the user for an integer between 5 and 10 (inclusive) and then reads in that many integer values and computes and prints the average of the numbers entered. For full credit, you should continue prompting the user if an invalid value is entered and the program should never throw an exception.

VII. (10 points, 5 points each part) Consider the following two methods:

```
public static void mystery1(int x, int y) {
  mystery2(x+y);
  if (x < y) {
     System.out.println(x);
     mystery2(2*x);
  }
  else {
    System.out.println(y);
      mystery2(7*y);
  }
}
public static void mystery2(int z) {
  if (z < 10 && z > 5)
    System.out.print("z");
  else
    System.out.print(z+1);
  System.out.println(z-2);
}
```

What is printed by each of the following:

a) `mystery1(2,4);`

b) `mystery1(7,4);`

VIII. (10 points) Write a method called `printMaxElement` that takes an array of integers as a parameter and prints the elements of the array followed by the maximum of these elements. You should complete this method without calling `Arrays.toString` and the output should have for format shown in the examples below.

```
int[ ] a = {-2, 5, 3, 6, 2};
int[ ] b = {6, 7, 4};
printMaxElement(a);
printMaxElement(b);
```

should print:

```
The maximum of -2, 5, 3, 6, 2 is 6
The maximum of 6, 7, 4 is 7
```

Hint: This will require a fencepost loop.

IX. (10 points) What does the following program print out? Mark your answer clearly.

```
import java.util.*;
public class MysteryComputation {
   public static void main(String[] args) {
      int[ ] c = {-4,-2,0,2,4,};
      int mysteryVar=mystery(c);
      System.out.println(Arrays.toString(c));
      System.out.println(mysteryVar);
   }
   public static int mystery(int[ ] b ){
      int var=0;
      for (int i=0; i<b.length; i++){
         b[i] = b[i]+2;
      }
      for (int i=0; i<b.length; i++){
         var=var + b[i];
      }
      return var;
   }
}
```

X. (10 points) Write a method, `numberDaysInMonth`, that takes two integer parameters, a month of the year (with January being month 1) and a year. If the month is in the range 1-12 and the year is greater than 0, it returns the number of days in the month for that year. Otherwise it returns 0. There are always 30 days in September, April, June, and November. These are the months numbered 9, 4, 6, and 11. All of the other months have 31 days, except February (month 2). February has 28 days unless it is a leap year when it has 29 days. A year is a leap year if it is divisible by 4, unless the year is also divisible by 100. Years that are divisible by 100 are leap years only when they are also divisible by 400. The year 2000 was a leap year, but the year 1900 was not.

Table 3.2 Useful Static Methods in the `Math` Class

Method	Description	Example
`abs`	absolute value	`Math.abs(-308)` returns `308`
`ceil`	ceiling (rounds upward)	`Math.ceil(2.13)` returns `3.0`
`floor`	floor (rounds downward)	`Math.floor(2.93)` returns `2.0`
`max`	maximum of two values	`Math.max(45, 207)` returns `207`
`min`	minimum of two values	`Math.min(3.8, 2.75)` returns `2.75`
`pow`	power (general exponentiation)	`Math.pow(3, 4)` returns `81.0`
`random`	random value	`Math.random()` returns a random `double` value k such that $0.0 \leq k < 1.0$
`round`	round real number to nearest integer	`Math.round(2.718)` returns `3`
`sqrt`	square root	`Math.sqrt(2)` returns `1.4142135623730951`

Table 3.3 Useful Methods of `String` Objects

Method	Description	Example (assuming `s` is `"hello"`)
`charAt(index)`	character at a specific index	`s.charAt(1)` returns `'e'`
`endsWith(text)`	whether or not the string ends with some text	`s.endsWith("llo")` returns `true`
`indexOf(text)`	index of a particular character or `String` (`-1` if not present)	`s.indexOf("o")` returns `4`
`length()`	number of characters in the string	`s.length()` returns `5`
`startsWith(text)`	whether or not the string starts with some text	`s.startsWith("hi")` returns `false`
`substring(start, stop)`	characters from start index to just before stop index	`s.substring(1, 3)` returns `"el"`
`toLowerCase()`	a new string with all lowercase letters	`s.toLowerCase()` returns `"hello"`
`toUpperCase()`	a new string with all uppercase letters	`s.toUpperCase()` returns `"HELLO"`

Useful Methods of `Scanner` Objects

Method	Description
`next()`	Reads and returns the next token as a `String`
`nextDouble()`	Reads and returns a `double` value
`nextInt()`	Reads and returns an `int` value
`nextLine()`	Reads and returns the next line of input as a `String`
`hasNext()`	Returns `true` if there is another token to be read
`hasNextDouble()`	Returns `true` if there is another token to be read and if it can be interpreted as a `double`
`hasNextInt()`	Returns `true` if there is another token to be read and if it can be interpreted as an `int`
`hasNextLine()`	Returns `true` if there is another line of input to be read

Table 4.5 Useful Methods of the `Character` Class

Method	Description	Example
`getNumericValue(ch)`	Converts a character that looks like a number into that number	`Character.getNumericValue('6')` returns `6`
`isDigit(ch)`	Whether or not the character is one of the digits `'0'` through `'9'`	`Character.isDigit('X')` returns `false`
`isLetter(ch)`	Whether or not the character is in the range `'a'` to `'z'` or `'A'` to `'Z'`	`Character.isLetter('f')` returns `true`
`isLowerCase(ch)`	Whether or not the character is a lowercase letter	`Character.isLowerCase('Q')` returns `false`
`isUpperCase(ch)`	Whether or not the character is an uppercase letter	`Character.isUpperCase('Q')` returns `true`
`toLowerCase(ch)`	The lowercase version of the given letter	`Character.toLowerCase('Q')` returns `'q'`
`toUpperCase(ch)`	The uppercase version of the given letter	`Character.toUpperCase('x')` returns `'X'`