Name:_____

CSE 1223: Exam II

Autumn 2016

Instructions:

- Do not open the exam before you are told to begin.
- This exam is closed book, closed notes.
- You may not use any calculators or any other kind of computing device on this exam.
- Direct all questions to the instructor and no one else
- You have 55 minutes to complete this exam. When time is called you must lay down your pencil and submit the exam.
- There are three parts to this exam containing questions worth a total of 100 points. Not all questions are worth the same number of points. You should scan the exam before starting.
- The space allotted for each questions should provide more than sufficient space to answer the question. If you find you are running out of space, you are probably on the wrong track.

Part	**Score**
I	/40
II	/34
III	/26
TOTAL	/100

In accordance with The Ohio State University Code of Student Conduct, I certify that:

- I have received no aid on this exam from any other person
- I have not given anyone aid on this exam
- I shall NOT discuss the contents of this exam with anyone who has not already taken this exam

Signed:_____

Part I – Coding (40 points)

1. (10 points) Complete the skeleton below for the method named `longest` that takes an array of Strings as input and returns the longest String in the array to the calling program. For example, if the array has the values `{"The", "Brown", "Fox"}` then the method will return the value `"Brown"`. In the case of ties your code can return ANY String with the longest length. For example if the array has the values `{"The", "Quick", "Brown", "Fox"}` you may return either `"Brown"` or `"Quick"`. You may assume that the arrays passed to the method will have a length of at least 1.

```
private static String longest(String[] array) {
     // your code goes here









}
```

2. (15 points) Fill in the skeleton below for a program that prompts the user for an integer. The program should then display a triangle of the numbers as shown below. The program should continue prompting for input until the user enters a negative number. The program should end with a Goodbye! message. See below for an example of the program in action. ***Make sure your code produces the same output as that given in the transcript below.*** Please note that a correct solution to this problem will NOT make use of any arrays.

   See the last page of the exam for some Scanner methods you might find useful. Here is a sample transcript of how the program should work. Input typed by the user is indicated by **bold** text:

```
Enter an integer: 3
3
33
333
Enter an integer: 2
2
22
Enter an integer: -1
Goodbye!
```

```
import java.util.Scanner;

public class Exam2B {

     public static void main(String[] args) {
          Scanner in = new Scanner(System.in);
          // your code goes here









     }

}
```

3. (15 points) Fill in the skeleton below for a method named `zeroOdds` that takes an array of integers as a parameter. The method should return a new array that has only the even values of the original array, with the odd values replaced by zeroes. For example, if the array `arr1 = {2,7,-1,8,5}`, then the call `zeroOdds(arr1)` should return the array `{2,0,0,8,0}`. Note that your code should work for arrays of length zero and that a zero length array with the odds zeroed out is another zero length array.

```
private static int[] zeroOdds(int[] array) {




}
```

Part II – Multiple Choice (34 points)

For each of the following questions, circle the single answer that BEST answers the question. (2 points each)

1. Consider the method header `public static String foo(int i)`. Which of the following could be a correct method call? (Note: in all of the examples below i is of type `int`, and `s` is of type `String`).
   a. `String foo(int i)`
   b. `String foo(i)`
   c. `s = foo(i)`
   d. `foo(String s)`

2. For the loop given below, what is the final value of `i` when the loop is finished?

```
int i = 1;
int arr = {2, 3, 5};
while (i < arr.length) {
    i = i + 1;
}
```

   a. 1
   b. 2
   c. 3
   d. 4
   e. You cannot tell just from the code segment provided
   f. The loop will never exit – it is an infinite loop

3. Suppose you have a problem where you are trying to find the larger of two numbers. Which ONE of the following Java programming concepts would be the best fit to use to solve this problem?
   a. `while` loop
   b. `if-else`
   c. `String` length
   d. None of the above – in the space below explain how you can choose the larger of two values without using any of the above concepts

4. A Java program that is supposed to compute the average of an array of numbers tries to access a number at an index larger than the length of the array. This is an example of what kind of error:
    a. Runtime error
    b. Syntax error
    c. Array Too Small error
    d. None of the above

5. What value does the variable `msg` have at the end of this segment of code?

```
int value = 31;
String msg = "";
if (value < 50) {
     msg = "Under 50!";
}
else if (value > 30) {
     msg = "Over 30!";
}
if (value < 25) {
     msg = "Under 25!";
}
```

    a. ""
    b. "Under 25!"
    c. "Over 30!"
    d. "Under 50!"
    e. You cannot tell from the code provided

6. For the loop given below, how many times will the boolean expression `(i < 2)` be evaluated?

```
int i = 0;
while (i < 2) {
    i = i + 1;
}
```

    a. 0
    b. 1
    c. 2
    d. 3
    e. 4
    f. The loop will never exit – it is an infinite loop

7. If you declare an array as `char[] cs = {'4','3','2','1'}`, then the array element `cs[3]` contains what value?

a. ‘1’
b. ‘2’
c. ‘3’
d. ‘4’
e. It is impossible to tell

8. What will be printed by the println statement at the end of this code segment?

```
int i = 2;
while ( i < 3 ) {
  String msg = "Value: "+i;
  i = i + 1;
}
System.out.println(msg);
```

a. “Value: 0”
b. “Value: 1”
c. “Value: 2”
d. “Value: 3”
e. None of the above – the above segment has an error in it. In the space below give a *brief* explanation of what the error is:

Each of the following questions may have multiple answers. Circle ALL responses that are answers to these questions (3 points each)

9. The method declared with the header `public static double oneThirdOf(int i)` returns:
   a. An integer value
   b. A double value
   c. The value of the parameter `i` divided by 3
   d. No value
   e. You cannot tell from the information provided

10. If x, y and z are **`boolean`** variables, for what values of x, y and z does the expression `((!x && y) || z)` evaluate to **true**? (Mark ALL that apply):
    a. `x=false, y=true, z=false`

b. `x=true, y=false, z=false`
c. `x=false, y=false, z=true`
d. `x=true, y=true, z=true`
e. The expression can be true, but not for any of the assigned values listed above
f. The expression cannot be true, no matter how you assign values to x, y and z

11. If `i`, `j` and `k` are **`int`** variables, for what values of `i`,`j` and `k` does the expression `((( i < j) || (i < k)) && (j * k > 10))` evaluate to **true**? (Mark ALL that apply):
    a. `i=8, j=9, k=1`
    b. `i=4, j=3, k=5`
    c. `i=8, j=10, k=1`
    d. `i=11, j=11, k=9`
    e. None of the above assignments of i, j and k make the expression true

12. The following segment of code provides examples of which of the following (mark ALL that apply):

```
i = 1024;
if (i % 2 == 0) {
     i = i / 4;
}
```

    a. boolean expression
    b. integer expression
    c. looping
    d. branching
    e. None of the above

13. Which of the following method headers are examples of *procedures*? (Mark ALL that apply)
    a. `private static void barfoo(int x, String s)`
    b. `private static String foobar()`
    c. `private static void foo(double x, char c)`
    d. `private static int bar(String x)`
    e. None of the above are examples of functions

14. Which of the following facts about arrays and ArrayLists are **false**? (Mark ALL that apply)
    a. Items in an array must all have the same type
    b. The size of an array does not need to be known when it is created
    c. The size of an ArrayList must be known when it is created

d. ArrayLists can never be empty
e. None of the above are false

Part III – Short Answer (26 points)

Consider the following code segments:

```
public static void main(String[] args) {.....................(1)
  int enigma = 12;...........................................(2)
  int mystery = 15;..........................................(3)
  int riddle = 30;...........................................(4)
  while(riddle>mystery) {....................................(5)
    int result = myMethod(mystery, riddle);..................(6)
    System.out.println(result);..............................(7)
    riddle--;.................................................(8)
    mystery++;................................................(9)
  }..........................................................(10)
  System.out.println("DONE!");...............................(11)
}............................................................(12)

private static int myMethod(int enigma, int mystery) {.......(13)
  int riddle = enigma * mystery;.............................(14)
  return riddle;.............................................(15)
}............................................................(16)
```

1. What are the line numbers of the lines that are in the scope of the variable `result` declared at line 6? (2 points)

2. What are the line numbers of the lines that are in the scope of the variable `riddle` declared at line 4? (2 points)

3. What are the line numbers of the lines that are in the scope of the variable `riddle` declared at line 14? (2points)

4. What are the line numbers of the lines that are in the scope of the variable `enigma` declared at line 2? (2points)

Trace the effect of the call to the method enigma below by tracing the execution of the method body of enigma for this call. (6 points).

Code	State (Variable Values)
	i = 5 j = 6
`k = enigma(i, j);`	
	i = j = k =

Tracing table for `enigma`:

Code	State (Variable Values)
`private static int enigma(int x, int y){`	
	x = y =
`  y = x / 2;`	
	x = y =
`  x = y + y*2;`	
	x = y =
`  int j = x - y;`	
	x = y = j =
`  return j;`	
`}`	

Consider the following code segment and answer the questions below. HINT: Doing question 9 first might help you with questions 6 – 8 if you are stuck.

```
int i;
int j;
int[] arr = {6, 4, 7, 2, 1};
for (i = arr.length - 1; i > 1; i--)
{
   System.out.println("Hello World!");
   j = arr[i];
}
```

6. How many times will the above code print the sentence “Hello World!”? (3 points)

7. What is the value of the integer variable *i* after executing the for loop? (3 points)

8. What is the value of the integer variable j after executing the for loop? (3 points)

9. Convert the for loop above to a while loop. You need to ensure that your code prints the sentence “Hello World!” the same number of times and ends with the same value for the integer value for the variables *i* and *j*. (3 points)

```
int i;
int j;
int[] arr = {6, 4, 7, 2, 1};
```

Summary of Possibly Useful Methods

String methods

- `int s.length();` // returns the length of the String *s*
- `char s.charAt(int pos);` // returns the character at position *pos* in the String *s*
- `int s.indexOf(String s2);` // returns the index of the first occurrence of the String *s2* in the String *s*
- `String s.substring(int start,int end);` // returns the substring of the String *s* starting at start and ending at end -1
- `boolean s.equals(String s2);` // returns true if s is the same sequence of characters as s2, otherwise false

Scanner methods

- `String s.nextLine();` // returns the next line from the Scanner *s*
- `int s.nextInt();` // returns the next integer from the Scanner *s*
- `double s.nextDouble();` // returns the next double value from the Scanner *s*

The rest of this page may be used as additional workspace as needed