

Name: \_\_\_\_\_

## CSE 1223: Exam I

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.

<u>Part</u>	<u>Score</u>
I	/30
II	/41
III	/29
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 (30 points)

1. (13 points) Complete the following program skeleton for the program Exam1A given below. This program should ask the user to enter a String. It should report the length of the String. Then it should ask the user to enter a second String and report the length of that String. It should then concatenate the two strings together with the longer string coming first and the shorter string coming second and display that to the screen. *Make sure your code produces the same output as that given in the transcript below.*

**You should get the input using the *Scanner* method `nextLine()`.** See the last page of the exam for reminders of what these Scanner methods and String methods do. Here is a sample transcript of how the program should work. Input typed by the user is indicated by **bold** text:

```
Enter a string: Hello
String has length: 5
Enter another string: Goodbye
String has length: 7
Concatenated together: GoodbyeHello
```

```
import java.util.Scanner;
```

```
public class Exam1A {
```

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

```
    }
```

```
}
```

2. (17 points) Complete the following program skeleton for the program Exam2B given below. **This program should ask the user to enter a String and should loop until the user enters an empty String.** If the user has entered at least one String, the program should report the longest and shortest strings entered by the user. *Make sure your code produces the same output as that given in the transcript below for the input given in the transcript below.*

You should get the input using the *Scanner* method `nextLine()`. See the last page of the exam for a description of the `nextLine()` method and some *String* methods that may be useful. Here is a sample transcript of how the program should work. Input typed by the user is indicated by **bold** text:

```
Enter a value: My
Enter a value: Good
Enter a value: Friend
Enter a value:
Longest String: Friend
Shortest String: My
```

```
import java.util.Scanner;
```

```
public class Exam2B {
```

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

```
    }
```

```
}
```

**Part II – Short Answer (41 points)**

3. Complete the tracing table below as discussed in class and in the course notes. **Make sure you trace every variable through every statement that is executed and indicate any iterations where code is skipped by drawing a line or leaving a blank.** (10 points)

<u>Code</u>	<u>State (Variable Values)</u>
	x = 3
<code>while (x % 2 != 0) {</code>	
	x =
<code>  if ( x &gt;= 5 ) {</code>	
	x =
<code>    x = x * 2;</code>	
	x =
<code>  } else {</code>	
	x =
<code>    x = x + 2;</code>	
	x =
<code>  }</code>	
	x =
<code>}</code>	
	x =

4. Consider the two code segments below (a and b), where  $x$  and  $y$  are `int` variables:

```
a)  if (x > y) {
      x = x / 2;
    }
    else {
      y = y + 3;
    }
```

```
b)  if (x > y) {
      x = x / 2;
    }
    if (x <= y) {
      y = y + 3;
    }
```

Will these two code segments *always* produce the same assignments for  $x$  and  $y$  in the end, or are there certain values of  $x$  and/or  $y$  where they will behave differently? If you believe that they will behave the same way, provide a convincing argument in 1-2 sentences to explain why you believe that this is so. If you believe that they will behave differently, provide at least one example of values for  $x$  and  $y$  where they behave differently and describe the difference (3 points).

Given the following declarations, determine the **type** (e.g. `int`, `boolean`, `String`, `double` or `char`) and **value** of each of the expressions listed below. For example, if the expression were `6+3` the **type** would be `int` and the **value** would be `9` (2 points each).

```
double pi = 3.14;
```

```
String foo = "Autumn2016";
```

```
int i = 7, j = 3;
```

```
boolean x = true, y = false;
```

5. `i < 6`

8. `foo.length()`

6. `x || y`

9. `foo.charAt(3)`

7. `j * pi`

10. `i % j`

Consider the following code segment where x is a variable of type int:

```
if (x > 55 ) {
    if (x < 80) {
        System.out.println("Fruit Custard");
    }
    else {
        System.out.println("Brownies");
    }
}
else if (x > 22) {
    if (x < 60) {
        System.out.println("Espresso Cake");
    }
    else {
        System.out.println("Apple pie");
    }
}
else if (x < 25) {
    System.out.println("Dark Chocolate Cake");
}
else { System.out.println("Cherry Pie"); }
```

For each of the following output values, give the range of values for the variable x that will cause the segment to display the given String. For example, if anything between 0 and 10 including both 10 and zero will produce the string, your answer should be  $0 \leq x \leq 10$ . If a value can never be produced by this code, give a short single sentence explanation of why not (2 points each).

11. "Apple pie"

12. "Brownies"

13. "Cherry Pie"

14. "Dark Chocolate Cake"

15. "Espresso Cake"

16. "Fruit Custard"

For the questions below, use the following variable declarations. As a reminder: *strictly* larger or smaller than  $x$  means that  $x$  is not included, and *strictly* between  $x$  and  $y$  means that the value is between  $x$  and  $y$  but does not include  $x$  and  $y$  (2 points each):

```
int count, max, min;
```

```
boolean stop, pause, end;
```

```
String foo, bar;
```

17. Write a **while** condition for a loop that runs as long as `max` is strictly greater than `min` and also less than the length of the string `bar`.

```
while ( _____ )
```

18. Write a while condition for a loop that runs as long as `stop` is false, `min` is strictly less than 6, and the character at position 3 in the String `bar` is not a 'q' .

```
while ( _____ )
```

### Part III –Multiple Choice (29 points)

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

19. Which of the following is the best definition of a *compiler*?

- a. a device for executing programs
- b. a program that translates source code into machine code
- c. a general term for a set of instructions for solving a problem
- d. a program used to edit source code

20. What value does the variable `foo` have at the end of this code segment?

```
int bar = 2;
```

```
int foo = 3 / bar;
```

- a. 0
- b. 0.5
- c. 1
- d. 1.5
- e. None of the above

21. What value does the variable `bar` have at the end of this code segment?

```
int foo = 12;
int bar = 0;
if (foo % 4 < 10) {
    bar = 1;
}
else {
    bar = 3;
}
if (foo % 3 < 10) {
    bar = 2;
}
```

- a. 0
- b. 1
- c. 2
- d. 3
- e. You cannot tell from the code segment provided

22. What value does the variable `i` have at the end of this segment of code?

```
String foo = "Au2016";
int i = foo.indexOf("2");
```

- a. 0
- b. 1
- c. 2
- d. 3
- e. 4
- f. You cannot tell from the code segment provided

23. Suppose `bar` has a value of 3. What value does the variable `foo` have after the following line of code executes?

```
boolean foo = (bar * 2) / 3 > 8;
```

- a. 2
- b. 9
- c. true
- d. false
- e. None of the above

24. What value does the variable `foo` have at the end of this segment of code?

```
String bar = "Au2016";  
String foo = bar.substring(2,5);
```

- a. "2016"
- b. "u20"
- c. "201"
- d. "016"
- e. None of the above

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

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

26. In the following segment of code, what concept is `c` an example of?

```
String s = "hello";  
char c = s.charAt(2);
```

- a. a type
- b. a method
- c. a variable
- d. a program
- e. None of the above

27. Which of the following is the best description of a *loop*?

- a. Chooses between two blocks of code to execute based on the result of a boolean expression
- b. Organizes a block of code into a group of statements that can be called using a name and parameters
- c. Executes a block of code possibly multiple times
- d. Assigns a value to a variable

28. The following code segment contains an error. What *type* of error is it?

```
String foo = "Hello";  
char c = foo.charAt(5);
```

- a. Syntax error
- b. Runtime error
- c. Division by zero error
- d. Logic error

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

29. If  $x$  and  $y$  are **boolean** variables, for what values of  $x$  and  $y$  does the expression  $(y \ || \ !x)$  evaluate to **false**? (Mark ALL that apply):

- a.  $x=\mathbf{true}$ ,  $y=\mathbf{true}$
- b.  $x=\mathbf{true}$ ,  $y=\mathbf{false}$
- c.  $x=\mathbf{false}$ ,  $y=\mathbf{true}$
- d.  $x=\mathbf{false}$ ,  $y=\mathbf{false}$
- e. The expression can never be true, regardless of what values  $x$  and  $y$  have

30. If  $i$  and  $j$  are **int** variables, for what values of  $i$  and  $j$  does the expression  $(i + j < 12 \ \&\& \ i > j)$  evaluate to **true**? (Mark ALL that apply):

- a.  $i=6$ ,  $j=6$
- b.  $i=3$ ,  $j=2$
- c.  $i=2$ ,  $j=9$
- d.  $i=8$ ,  $j=5$
- e. None of the above assignments of  $i$  and  $j$  make the expression true

31. The following segment provides example(s) of which of the following (Mark ALL that apply):

```
while (foo + bar <= 10) {  
    foo = foo * bar;  
}
```

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

## 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`

### 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**