

CS 102: Section 02

Data Structures and Algorithms

Spring 2017 Midterm

Name: _____

Instructions:

1. This is an open notes exam.
2. BUDGET YOUR TIME. This is a 120 minute exam. You can spend about 2 minutes per point (e.g., 20 minutes on a 10 point question). Don't spend too much time on any one question.
3. Use a pencil to write this exam. If you don't have one, ask your professor to loan you one.
4. Each question indicates the total number of points.
5. Please don't hesitate to ask questions during the exam.
6. Good luck!!!

Question	Points	
	received	possible
1. Short answer and reading and understanding code		10
2. Question about exception handling		5
3. Writing code by example		10
4. Writing a method in a structurally recursive list		5
5. Java Inheritance Hierarchy		10
6. Short answer and True/False		25
Total		65

1. (10 points) Short answer questions:

- (a) (1 point) What is the last index number in an instantiated array of n elements?

- (b) (2 points) In Java, classes have 2 fundamentally different purposes. Name or describe each of these purposes.

- (c) (2 points) Name two different **classes** we have used for writing output.

- (d) (4 points) Give one example of the possible outputs for each of the following 2 code segments (assume they are each part of the main method in 2 different classes) and explain the purpose of each segment at areas indicated below:

```
1. int x = 0, y = 0;
2. do {
3.     x = (int)(10*Math.random() + 1);
4.     y = (int)(10*Math.random() + 1);
5. } while (x != y);
6. System.out.println(x + " " + y);
```

Example output:

Purpose:

```
1. int x,y;
2. x = y = 0;
3. while (x == y) {
4.     x = (int)(10*Math.random() + 1);
5.     y = (int)(10*Math.random() + 1);
6. }
7. System.out.println(x + " " + y);
```

Example output:

Purpose:

2. (5 points) The following question concerns the class definition shown below:

```
1. public class RandomArray {
2.     public static void main(String[] args) {
3.         int size = 3;
4.         int[] inches = new int[size];
5.         try{
6.             for (int i = 0; i <= size; i++){
7.                 inches[i] = (int)(20 * Math.random() + 1);
8.                 System.out.println(""+inches[i]+" inches of rain.");
9.             }
10.        }catch(ArrayIndexOutOfBoundsException aio) {
11.            System.out.println("Inside catch block.");
12.        }
13.        System.out.println("All's well.");
14.    }
15. }
16. }
```

From the 4 sequences of output below, circle the possible outputs of this code and explain why the non-circled outputs are not possible output:

19 inches of rain.	14 inches of rain.	18 inches of rain.	12 inches of rain.
8 inches of rain.	24 inches of rain.	5 inches of rain.	10 inches of rain.
6 inches of rain.	2 inches of rain.	10 inches of rain.	11 inches of rain.
Inside catch block.	Inside catch block.	All's well.	Inside catch block.
All's well.	All's well.		All's well.

3. (10 points) Consider the code for class DigitSum given below:

```
1. import java.util.*;
2. public class DigitSum{
3.     public static void main(String[] args) {
4.         Scanner scan = new Scanner(System.in);
5.         System.out.println("This program sums the digits in an integer.");
6.         System.out.println("Enter a positive integer: ");
7.         int n = scan.nextInt();
8.         int nSave = n;
9.         int digitSum = 0;
10.        while (n > 0) {
11.            digitSum += n % 10;
12.            n /= 10;
13.        } // end while
14.        System.out.println("The sum of the digits in " +
15.                            nSave + " is " + digitSum);
16.    } // end main } // end class

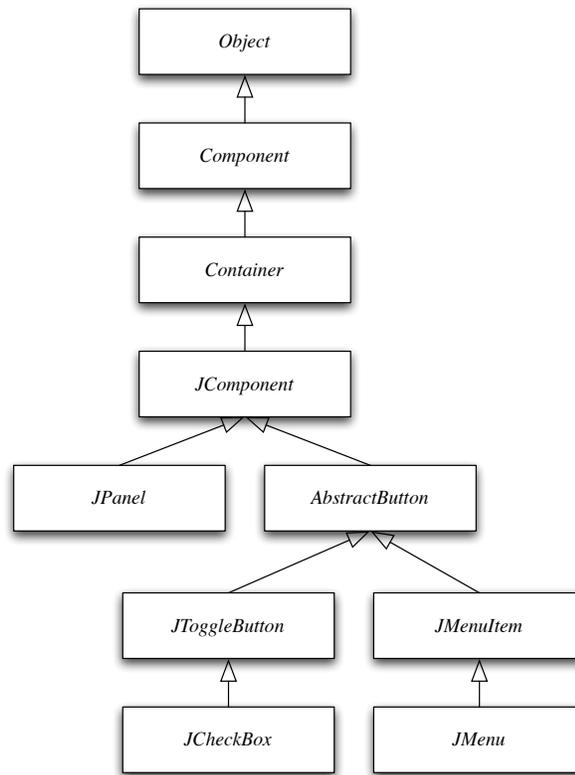
/* Sample output */
This program sums the digits in an integer.
Enter a positive integer: 161
The sum of the digits in 161 is 8.
```

Using DigitSum as a model, fill in the blank part of the class named DigitCount that counts the number of digits in the number the user enters. You need only fill in the body of the main method shown below. Sample output is given below the class definition. If any of the lines in DigitSum can be used verbatim in DigitCount, write “Same” to the right of each line that is the same as it is in DigitSum. For example, “Same” is written to the right of line number 4 below. Lines 1 and 3 should also be marked “Same”, but they are given for clarity. You may also choose to write every line.

```
1. import java.util.*;
2. public class DigitCount {
3.     public static void main(String[] args) {
4.         Same
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.    }
15. }
/* Sample output */
This program counts the number of digits in an integer.

Enter a positive integer: 1861
The number of digits in 1861 is 4
```


5. (10 points) Refer to the Java Inheritance hierarchy shown below to answer the following questions.



- (a) (3 points) List or mark all classes in the hierarchy in which a method to be called on a *JCheckBox* object could be defined. *Explain your answer.*
- (b) (3 points) List or mark all the classes that inherit code from class *JComponent*.
- (c) (4 points) Suppose you were asked to implement the *JCheckBox* and *JMenu* classes as Java programs. Write class header lines (signatures) for each of the *JCheckBox* and *JMenu* classes. Be sure that each class header reflects the inheritance relationship shown in the figure above.

6. (25 points) Short answer and True/False. Explaining your answers may factor into the correctness.
- (a) (2 points) What are *instance variables* and *instance methods*? What distinguishes instance members from non-instance members inside a class definition?

 - (b) (2 points) How are *instance methods* called from within the main method of the same class?

 - (c) (2 points) What is the purpose of a constructor? Write a constructor signature for a class called `PersonData` that has 2 instance variables: a String **name** and an int **age**.

 - (d) (1 point) What is the purpose of the keyword *new*?

 - (e) (2 points) List two keywords that could be used to create inheritance (“is-a”) relationships between files `x.java` and `y.java`.

 - (f) (2 points) What is an *interface*?

(g) (2 points) Suppose you came across a class with the header

```
public class Car implements Vehicle
```

Circle the statements in the group of 4 below that would create a new object of type Vehicle.

```
Vehicle myCar = new Vehicle();
```

```
Car myExtraCar = new Car();
```

```
Car myOtherCar = new Vehicle();
```

```
Vehicle anotherCar = new Car();
```

- (h) (1 point) True or False? When passing parameters, the name of the argument must match the name of the parameter.
- (i) (1 point) True or False? The package `java.lang` must be imported in every java class you write.
- (j) (1 point) True or False? Every package p besides `java.lang` must be imported in order to use the classes defined in package p .
- (k) (1 point) True or False? When passing parameters, the type of the argument must match the type of the parameter.
- (l) (1 point) True or False? Syntax errors always occur during runtime.
- (m) (2 points) Name 2 keywords that may be used to avoid program failure when exceptions occur during runtime.

(n) (5 points) Consider the following boolean expression, where variable y is a positive `int`:

$$(y \% 2 == 0) \ \&\& \ (y \% 7 == 0)$$

For what value(s) of y does this expression evaluate to **false**?

For what value(s) of y does this expression evaluate to **true**?

For what value(s) of y does this expression short-circuit?