Name:

CS-1110 Practice Final

- You are allowed one 2-sided note sheet
- No sharing of materials
- Do work for a problem only on the designated sheet(s) for the problem
- Show all work so partial credit can be given
- You must turn in your note sheet with your exam

Instructions: Please read each question carefully and follow the instructions exactly as written. Some questions have more that one part. Show all of your work in the spaces provided.

1) True/False (T or F), Multiple Choice

____ A **do/while** loop is always executed at least once.

____ The correct way to exit a loop when the counter has not reached its exit value is to use a **break** statement.

____ Arrays may only store primitive values, such as **int**s and **double**s.

____ Constructors may not be overloaded

____ Comparing two **int** values should be done using the .equals() method.

____ A subclass has the same or more capabilities than its parent class

____ An abstract class must have at least one abstract method.

____ A finally block will run only after a catch block handles an exception.

____ The **Object** class is the superclass of all classes

Name:

2) Short answer. Be clear and concise.

(a) When should you use static methods or variables instead of nonstatic.

(b) What is the difference between an Interface and an Abstract Class in Java?

(c) What advantage can using Try-With-Resources give you over a standard try block? What interface does any object declared in the Try-With-Resources header need to implement?

Name:

(d) In the code below, assign the appropriate `format` strings to each line to achieve the following output:

```
86753098675309
3.14159
2934.53 006
```

```
public class PrintfQuestion {
    public static void main(String[] args) {


        System.out.printf("_________", "86753098675309");
        System.out.printf("_________", Math.PI);
        System.out.printf("_______________", 2934.5321132, 6);
    }
}
```

(e) If a = 3 and b = 1, what is the value of c?

```
c = a % b == 0 ? a + b : a - b;
```

(f) List at least **three** methods of an ArrayList.

Name:

(g) What does the .equals method need to check for when comparing objects? (There are 4)

(h) Change this for each loop to a for loop:

```
public class NumbersAreFun {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3, 4, 5};
        for (int number : numbers) {
            System.out.println(number);
        }
    }
}
```

Name:

3) (21 points) Multiple choice: Answer the following questions by circling the letter corresponding to the **best** choice of the options given.

(a) The result of 23 / 2 + 6 - 3.4 is

A. -1.6 B. 4 C. 13.6 D. 13 E. None of the above

(b) Which of the following identifiers follows the coding standard for a variable?

A. myidentifier B. my_identifier C. MyIdentifier D. myIdentifier

(c) Which of the following conditions would correctly end a loop when a certain value **a** is within the range of 1.0 and 15.0?

A. a < 1.0 && a > 15.0
B. a >= 1.0 && a <= 15.0
C. a >= 1.0 || a <= 15.0
D. a > 1.0 || a < 15.0

(d) The following code is an example of _____.

```
int cents = (int) (amountInDollars * 100);
```

A. Syntax Error
B. Typecasting
C. Logic Error
D. Self-Documenting Code
E. Runtime Error

(e) _____ is written by the programmer in an IDE and later compiled down to Bytecode by the compiler.

A. Algorithm
B. Source Code
C. Java Virtual Machine
D. Machine Code
E. Pseudocode

Name:

(f) What is the value of the variable `prod` after the following Java code segment has been executed?

```
int prod = 1;
for(int count = 3; count < 6; count++) {
    prod *= count;
}
```

A. 3 B. 60 C. 360 D. 840

(g) Given `String x = "hello";` what is printed by `System.out.print(x.indexOf('l'));`?

A. 3
B. 4
C. An error message
D. 2
E. 23

(h) Select the letter that best classifies the following code:

```
public class FinalExam {
    public static void main(String[] args) {
        int[] a = new int[6];
        int i = 0;
        while(i < 6) {
            a[i] = i * i;
            System.out.println(a[i]);
        }
        i++;
  }
}
```

A. Won't compile, and therefore won't run
B. Will run, but may not terminate
C. Will run with runtime errors
D. Will run with no runtime errors

(i) Which is a possible value for `x` if the output of the following Java segment is `Boo!`?

```
if(x % 2 == 1 && x < 7) {
    System.out.println("Boo!");
} else {
    System.out.println("Eek!");
}
```

A. 4 B. 9 C. -3 D. 5 E. 7

Name:

```
public static void main(String[] args) {
  try {
     Scanner in = new Scanner(System.in);
     System.out.println("Hello, World!");
          int x = 6;
     x = Integer.parseInt(in.nextLine());
     int y = Integer.parseInt(in.nextLine());

     System.out.println(x / y + " is the answer.");
  } catch (NumberFormatException e) {
     System.out.println("Number?");
  } catch(IllegalArgumentException e) {
     System.out.println("Words?");
  }
}
```

(j) Show the output of the above code given the following input:

```
"7", "3"


"x", "2"
```

4) (20 points) Use the next page to complete the following Java program to compute a mail-order bill for each customer of a mail-order house. This mail-order house sells just two types of T-shirts: T-shirt style 1 costs $4.50 per item and T-shirt style 2 costs $9.50 per item. In addition, if the total order is less than $30, there is a $3.50 surcharge. For each customer, the input to the program is a customer ID (a string), a style number (1 or 2), and a quantity ordered. The output for each customer will be the ID and the amount billed. For example, if the input line was

```
DOE123 1 5
```

the output would be

```
Customer with ID: DOE123 owes $26
```

since 5 x 4.50 = 22.50 and 22.50 + 3.50 = 26. Note that each order is for just one style of T-shirt; no order includes both style 1 and style 2.

The program must read its input from `System.in` and write the output to `System.out`. Read multiple T-shirt orders, one order per line. Stop reading when the T-shirt style is 0. You are not required prompt the user for input (i.e., you do not need a print statement asking for the order) or format the number of decimals in the output. You may assume that user input is always valid.

Name:

```
2 shirts- total less than $30 sir charge
public class ProcessOrders {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
```

Name:

5) Create a UML: Suppose you have a Vector class representing mathematical vectors in 3D space. Implement a driver program to perform vector operations and display the results

Your Vector class should include the following functionalities:

1. Constructors to initialize the vector components.
2. Methods to perform vector addition, subtraction, scalar multiplication, dot product, and cross product.
3. Override the toString() method to display the vector in a readable format.

Here's a sample driver program demonstrating some vector operations:

```
java
public class VectorDriver {
    public static void main(String[] args) {
        // Create two vectors
        Vector v1 = new Vector(1, 2, 3);
        Vector v2 = new Vector(4, 5, 6);

        // Perform vector operations
        Vector sum = v1.add(v2);
        Vector difference = v1.subtract(v2);
        double scalarProduct = v1.scalarProduct(v2);
        Vector crossProduct = v1.crossProduct(v2);

        // Display the results
        System.out.println("Vector 1: " + v1);
        System.out.println("Vector 2: " + v2);
        System.out.println("Sum: " + sum);
        System.out.println("Difference: " + difference);
        System.out.println("Scalar Product: " + scalarProduct);
        System.out.println("Cross Product: " + crossProduct);
    }
}
```

Create a UML

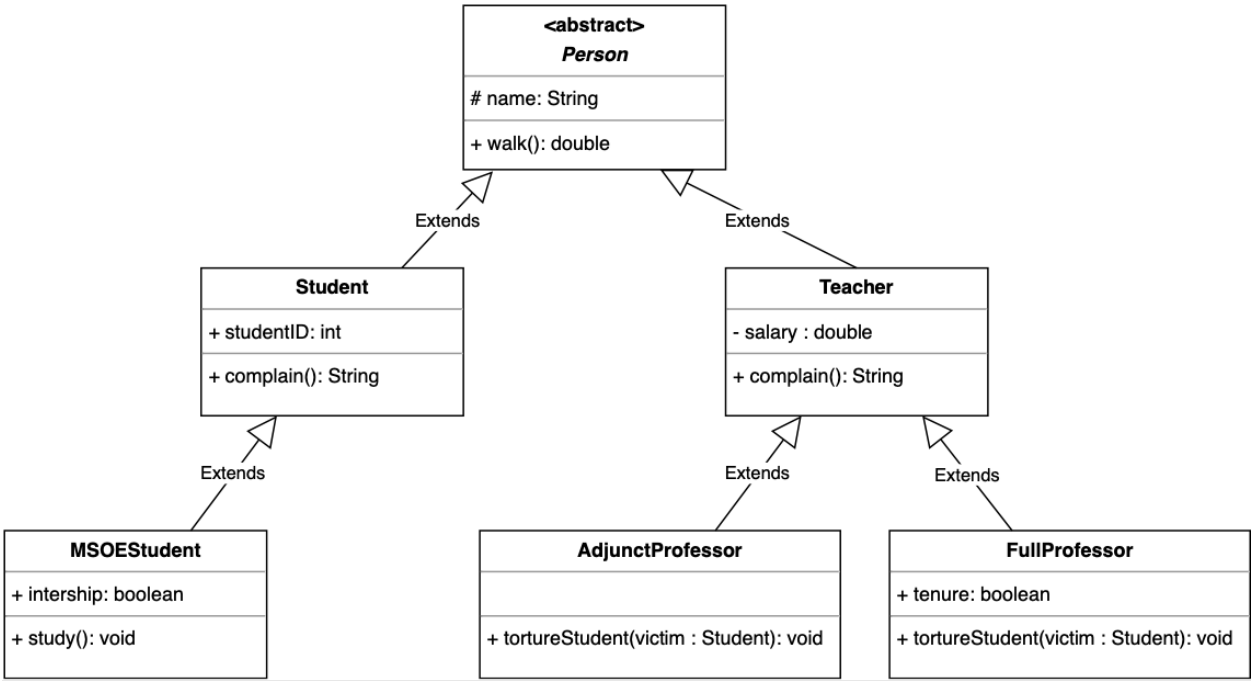
Name:

6) Complete the method below that will determine if a given String is a palindrome (a word that reads the same forwards and backwards, such as “racecar or tacocat). You may only use the following String methods in your solution: `charAt()`, `length()`, `substring()` though you may use other, non-String methods.

```
private static boolean isPalindrome(String word) {




}
```

Name:

7) Answer the following questions based on the UML diagram below.



a) Is the following code legal? Why or why not?

```
Person p = new Person();
```

b) What variables and methods will be directly available to an FullProfessor object?

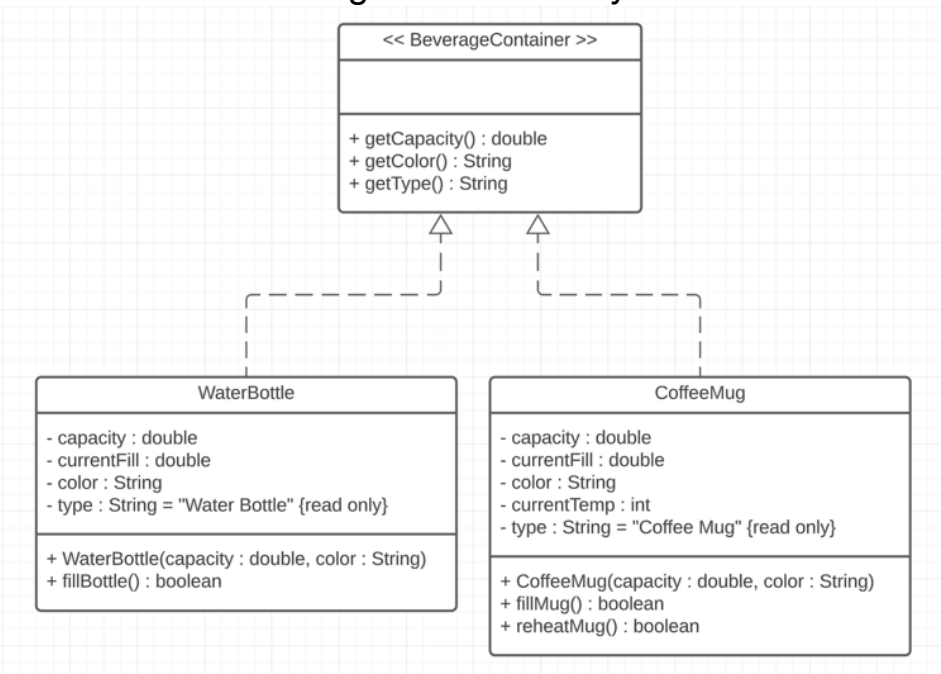
Name:

8) In the code below, identify and explain five unique violations of the coding standard used in this course. The same type of violation that occurs multiple times is considered a single violation. Only circle coding standard violations, not logic or coding errors.

```
/**Course
 * Practice Final Exam
 * Name: Sean Jones
 * Created: 10-31-19
 */
package exam2;

/*
 * A Class that calculates powers of 7 (because why not?)
 */
public class practiceexam2 {
 int b, e, x;

   /**
    * Constructor for the class
    * Precondition: must have a positive, non-zero exponential
    * value
    *
    * @param Exponent
    */
   public practiceexam2(int Exponent) {
      if(Exponent != 0) {

         this.e = Math.abs(Exponent);
      }
      else
      {
         this.e = 1;
      }
      this.b = 7;
      this.x = 1;
   }
   public int daMath() {
      for(int i=1;i<=e;i=i+1)  {
         x *= b;
      return x;
   }
}
```

Name:

9) Consider the following class hierarchy.



a) Implement the BeverageContainer Interface.

Name:

b) Implement the complete WaterBottle class. The functionality for fillBottle() should set the value of currentFill to capacity and return true. If currentFill already equals capacity the method should return false.

Name:

10) Write a program that asks a user for their first name, last name, age, and major. If the user enters either a negative number or something other than an integer for their age the program should re-ask for their age until an integer. Then the user should write this information to a file named <lowercase first letter of first name><lowercase last name>.txt. For example, if the user's name is John Doe, the output filename should be jdoe.txt. A sample run of the program is shown below. Text entered by the user is bolded for clarity. Recall that when writing files you need to handle the possibility of an IOException.

```
What is your first name? John
What is your last name? Doe
What is your age? Twenty One
Invalid input, please enter a positive integer value.
What is your age? -3
Invalid input, please enter a positive integer value.
What is your age? 21
What is your major? Underwater Basket Weaving
Information written to jdoe.txt
```

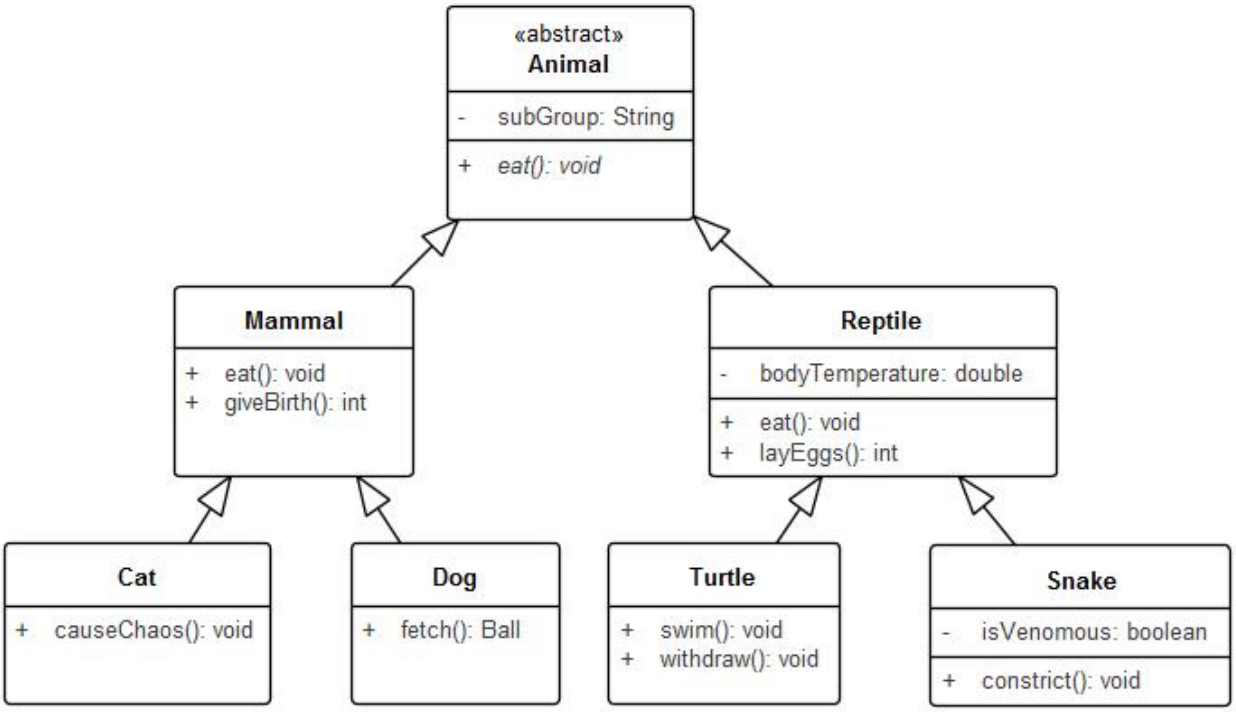
The information in the jdoe.txt file should be formatted as follows:

```
Name: John Doe
Age: 21
Major: Underwater Basket Weaving
```

Name:

Bonus:

Consider the following UML diagram when answering the parts below. Select the best answer.



1. Animal a = **new** Cat(); Explain Your Choice:
    1. Compiles, runs
    2. Does not compile
    3. Compiles, run-time error

2. Dog spot = **new** Mammal(); Explain Your Choice:
    1. Compiles, runs
    2. Does not compile
    3. Compiles, run-time error

3. For a Zoo class which has a List<Animal> attribute, the association between Zoo and Animal would be an example of
    1. Aggregation
    2. Dependency
    3. Inheritance

Name:

4. **Fill-in-the-blank**. Of the classes above _____ cannot be instantiated.

5. Which of the following snippets of code would you likely observe in the implementation of the UML diagram?
    1. **public class** Dog **implements** Mammal {....
    2. **public class** Reptile **extends** Snake {....
    3. **public class** Cat **extends** Mammal {....
    4. **public class** Turtle **extends** Reptile **implements** Animal {....
    5. **public class** Reptile **implements** Snake {....

6. An interface may contain the following (Choose all that apply, including choosing none):
    1. Constants
    2. Instance variables
    3. Method signatures (stubs/head)
    4. Constructors

7. If **A**, **B**, and **C** are classes, and **D** and **E** are interfaces, which one of the following is true?
    1. A class can have at most one super class and at most one interface such as **class** A **extends** B **implements** D …
    2. A class can have many super classes and many interfaces such as **class** A **extends** B, C **implements** D, E …
    3. A class can have at most one super class but many interfaces such as **class** A **extends** B **implements** D, E …
    4. A class can have many super classes but at most one interface such as **class** A **extends** B, C **implements** D …
    5. A class can have either one super class or one interface such as **class** A **extends** B … or **class** A **implements** D …