Full Name: _____

Student ID#: _____

CSE 1325 OBJECT-ORIENTED PROGRAMMING

PRACTICE #1 Exam #1 «---» 0 1 001 «---» Exam #1 PRACTICE #1

Instructions

1. Students are allowed pencils, erasers, and beverage only.
2. All books, bags, backpacks, phones, **smart watches**, and other electronics, etc. must be placed along the walls. **Silence all notifications.**
3. PRINT your name and student ID at the top of this page **and every coding sheet**, and verify that you have all pages.
4. **Read every question completely before you start to answer it.** If you have a question, please raise your hand. You may or may not get an answer, but it won't hurt to ask.
5. If you leave the room, you may not return.
6. You are required to SIGN and ABIDE BY the following Honor Pledge for each exam this semester.

Honor Pledge

On my honor, I pledge that I will not attempt to communicate with another student, view another student's work, or view any unauthorized notes or electronic devices during this exam. I understand that the professor and the CSE 1325 Course Curriculum Committee have zero tolerance for cheating of any kind, and that any violation of this pledge or the University honor code will result in an automatic grade of zero for the semester and referral to the Office of Student Conduct for scholastic dishonesty.

Student Signature: _____

Vocabulary

Write the word or phrase from the Words list below to the left of the definition that it best matches. Each word or phrase is used at most once, but some will not be used. {10 at 2 points each}

Vocabulary

Word	Definition
1	The class inheriting members
2	A method declared with no implementation
3	A program that runs in managed memory systems to free unreferenced memory
4	A declaration that also fully specifies the entity declared
5	An expression that, if false, indicates a program error
6	Specifying a general interface while hiding implementation details
7	A statement that introduces a name with an associated type into a scope
8	A named scope
9	A special class member that cleans up when an object is deleted (not supported by Java)
10	A block of memory associated with a symbolic name that contains a primitive data value or the address of an object instance

Word List

Abstract Class	Abstract Method	Abstraction	Algorithm	Assertion
Class	Constructor	Data Validation	Declaration	Definition
Destructor	Encapsulation	Exception	Field	Garbage Collector
Getter	Inheritance	Instance	Interface	Method
Multiple Inheritance	Namespace	Object-Oriented Programming	Operator	Override
Package	Primitive type	Reference Counter	Setter	Shadowing
Subclass	Superclass	UML	Validation Rules	Variable

Multiple Choice

Read the full question and every possible answer. Choose the one best answer for each question and write the corresponding letter in the blank next to the number. The image below clarifies the meaning of open and closed arrows and diamonds. {15 at 2 points each}

1. ____ **Java handles multiple inheritance using**

    A. Polymorphism

    B. Interfaces

    C. Abstract encapsulation

    D. Aggregation

2. ____ **To compare the value of int a and int b in Java, write**

    A. `a equals b`

    B. `a.equals(b)`

    C. `a == b`

    D. `a.==(b)`

3. ____ **The default destructor for class Account in Java is declared as**

    A. `public destructor Account()`

    B. `public static destructor Account();`

    C. `public void ~Account() {};`

    D. Java does not require or permit destructors

4. ____ **In the UML, an abstract method is shown with**

    A. bold

    B. underline

    C. italics

    D. strikethrough

5. ____ **To throw an exception for bad dates, write**

    A. `throw{Exception, "bad dates"};`

    B. `throw new Exception = "bad dates";`

    C. `throw "bad dates" as new Exception();`

    D. `throw new Exception("bad dates");`

6. ____ **Java is**

    A. Pass by const reference

    B. Pass by reference

    C. Pass by value

    D. Pass by pointer

7. ____ **The main method in class Gradebook can access which method in class Grade?**

    A. `protected double average()`

    B. `public double average()`

    C. `private double average()`

    D. All of the above

8. ____ **The Java default constructor for class BigInt can chain to an overloaded constructor using**

    A. `this.BigInt(42);`

    B. `this(42);`

    C. `BigInt(42);`

    D. Java does not support constructor chaining

9. ____ **To print "Hello, world!" to the console in Java, write**

    A. `printf("Hello, world!\n");`

    B. `System.console.out("Hello, world!\n");`

    C. `System.out.println("Hello, world!");`

    D. `std::cout << "Hello, world!\n";`

10. ____ **To invoke target "clean" in the local build.xml from the bash shell, type**

    A. `ant.clean()`

    B. `clean`

    C. `ant clean`

    D. `./clean`

11. ____ **In the UML, a class is represented by**

    A. A cloud

    B. A triangle

    C. An elipse

    D. A rectangle

12. ____ **In Java, we would convert a String s to all upper case letters using**

    A. `for(char c : s) c.toUpperCase();`

    B. `toUpperCase(s);`

    C. `String.toUpperCase(s);`

    D. `s.toUpperCase();`

13. ____ **One advantage of a Java ArrayList over a traditional array is**

    A. The ArrayList size can change dynamically while the array size is fixed when defined

    B. The ArrayList can be used with a for-each loop, but an array cannot

    C. The ArrayList is faster than an array

    D. They are the same - ArrayList is just the object-oriented name for a traditional array

14. ____ **Of the four arrows shown in the diagram above, the only one NOT a valid UML class relationship is**

    A. The closed arrow

    B. The open diamond

    C. The open arrow

    D. The closed diamond

15. ____ **Once class Animal has been declared, it may be instanced using**

    A. `Animal canis = new Animal("Fido");`

    B. `Animal canis("Fido");`

    C. `Animal canis = Animal("Fido");`

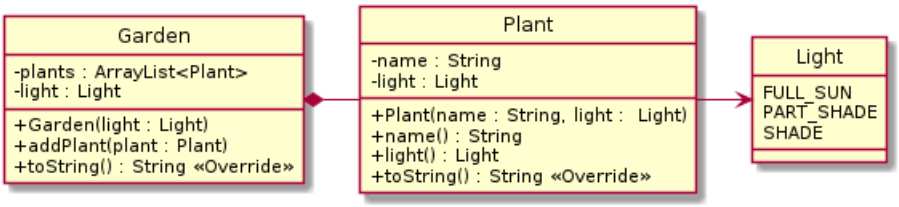
    D. All of these are valid

Free Response

Write the solutions to the Free Response questions in the provided space. Additional coding sheets are available on request. **Write your name and student ID on EVERY additional coding sheet you use that is not already stapled to this exam.**

Each question stands alone, with instructions on what to code from the class diagram. If you have trouble answering a question, simply move to the next question and come back later. **Assume all needed imports - don't code them.**

1. Consider the following class diagram.



a. {4 points} Given the above class diagram, **Write the public enum class Light.**

b. {7 points} Given the above class diagram, **write public class Plant**. Explicitly construct each field in the constructor. Methods name() and light() are getters. Override the toString method such that "Rose" in full sun would convert to the string "Rose (FULL_SUN)" and "Periwinkle" in shade would convert to "Periwinkle (SHADE)".

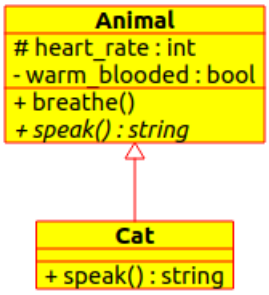
You do NOT need to write any code that uses this class – write the implementation file only.

c. {10 points} Given the class diagram on the previous page, write the implementation for **class Garden**. Explicitly construct field light and the ArrayList in the constructor. Method addPlant appends its parameter to ArrayList plants.

Override the toString method so that a Garden object with 2 plants - "Rose" in full sun and "Periwinkle" in shade - would convert to the string "Rose (FULL_SUN), Periwinkle (SHADE)".

You do NOT need to write any code that uses this class – write the implementation file only.

2. Given the class diagram, answer the questions below.



a. {3 points} Write a single line of code to instance class Animal as variable animal.
If that is not possible, briefly explain why.

b. {3 points} Define Cat's speak method (which prints "Meow" to the console) such that, if Animal.speak did not already exist, the javac compiler would issue an error.

c. {6 at 1 point each, 6 points total} Circle T for each true statement and F for each false statement.

T F Cat will inherit Animal's default constructor.

T F Cat.speak() can access Animal.heart_rate.

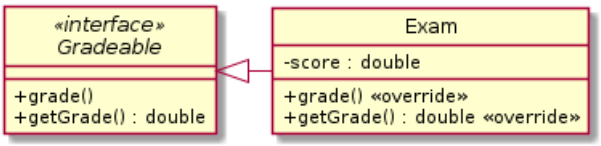
T F Cat.speak() can access Animal.warm_blooded.

T F `Cat c = new Cat(); c.breath();` will compile without errors.

T F The relationship of Cat to Animal is usually expressed in Java as `class Cat extends Animal`;

T F The return type for Animal.breath() is undefined

3. Consider this class diagram.



a. {3 points} Write Java interface Gradeable.

b. {4 points} Write class Exam that implements interface Gradeable above. Method grade assigns a random value to score between 40 and 100. Method getGrade returns score.

4. {10 points} Demonstrate error handling by adding code to the fragment below.

```
import java.util.Scanner;

public class Bank {
    public void deposit(double amount) {
         // Add code so that, if amount is not positive,
         //   an Illegal Argument Exception is thrown
         //   with the message "Non-positive deposit amount"



         balance += amount;
   }

  public static void main(String[] args) {

      // Add code so that, if the user enters a non-positive
      //   deposit amount below, catch the exception
      //   thrown in Bank.deposit(double) and print
      //   its message to the console's error output stream

       Bank bank = new Bank();
       Scanner in = new Scanner(System.in);
       double d = in.nextDouble();


          bank.deposit(d);




    private int balance;
}
```

Bonus: {+4 points} List and briefly describe two software licenses that you might choose for your own personal projects. This might be a specific license, or perhaps two of the six classes of licenses as discussed in the lecture.