Full Name: _____

Student ID#: _____

CSE 1325 OBJECT-ORIENTED PROGRAMMING

PRACTICE #3 Exam #1 «---» 1 1 001 «---» Exam #1 PRACTICE #3

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 additional pastel coding sheet**, and verify that you have all 10 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	A block of memory associated with a symbolic name that contains a primitive data value or the address of an object instance
2	Algorithmically enforceable constraints on the correctness, meaningfulness, and security of input data
3	A statement that introduces a name with an associated type into a scope
4	An instance of a class containing a set of encapsulated data and associated methods
5	Code for which specified assertions are guaranteed to be true (often, a class in which fields cannot change after instantiation)
6	An object created to represent an error or other unusual occurrence and then propagated via special mechanisms until caught by special handling code
7	A special class member that creates and initializes an object from the class
8	Ensuring that a program operates on clean, correct, and useful data
9	A style of programming focused on the use of classes and class hierarchies
10	A data type that can typically be handled directly by the underlying hardware

Word List

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

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. Note that "e.g." means "for example". {15 at 2 points each}

1. ____ **Which symbol (shown in double-quotes) denotes public visibility in a UML class diagram?**

    A. `` + `` (plus)

    B. `` – `` (dash)

    C. `` ~ `` (tilde)

    D. `` # `` (pound)

2. ____ **A custom Java exception LostSignalException may be created with**

    A. `class LostSignalException implements Exception`

    B. `exception LostSignalException`

    C. `enum LostSignalException {NONE, WEAK, DEGRADED}`

    D. `class LostSignalException extends Exception`

3. ____ **Refer to the image at the start of this section. Inheritance in UML is represented by a line with**

    A. An open arrow

    B. An open diamond

    C. A closed arrow

    D. A closed diamond

4. ____ **For class Bat to inherit from both Mammal and Flyer classes in Java we would write**

    A. `class Bat extends Mammal, extends Flyer`

    B. `class Bat extends Mammal, Flyer`

    C. `class Bat extends {Mammal, Flyer}`

    D. Java doesn't support multiple inheritance for classes

5. ____ **Given object car of type Car,** `System.out.print(car)` **will**

    A. print the result of Car's toString() method

    B. print the hash value of the car object

    C. print the memory address of the car object

    D. generate a compiler error

6. ____ **A final class**

    A. cannot be a subclass

    B. cannot be edited

    C. cannot be a superclass

    D. cannot be instanced

7. ____ **The heap memory allocated by the statement** `Foo foo = new Foo();` **is deallocated (released) when**

    A. foo goes out of scope and the garbage collector runs

    B. `foo.free()` is executed

    C. `free(foo);` is executed

    D. `delete foo;` is executed

8. ____ **Given a new file** `Tree.java` **in a clone of GitHub repository cse1325, which bash command will upload Tree.java to GitHub?**

    A. `git add Tree.java ; git commit -m "A tree!" ; git push`

    B. `git add push Tree.java -m "A tree!"`

    C. `git commit Tree.java  -m "A tree!" ; git add cse1325 ; git push`

    D. `git add commit Tree.java -m "A tree!"`

9. ____ **When the number of elements to be added is unknown, a good collection to use in Java is usually**

    A. an array

    B. `ArrayList`

    C. a custom class you write yourself

    D. separate variables

10. ____ **A UML class is represented by a rectangle with 3 panes. The middle pane contains**

    A. The executable members such as constructors and methods

    B. The fields

    C. The name of the class and an optional stereotype

    D. The middle pane is always empty

11. ____ **Class members with which level of visibility cannot be visible (accessible) in subclasses?**

    A. Protected

    B. Private

    C. Package-protected

    D. Public

12. ____ **When modifying text in Java, the fastest class for these changes is**

    A. StringBuilder

    B. char*

    C. QuickString

    D. String

13. ____ **Which statement is TRUE about String in Java?**

    A. A specific char within a String may be accessed with subscripts, e.g., `s[3]`

    B. In Java, `String` is another name for `char*`

    C. Two strings s1 and s2 may be concatenated (combined into one string) using `s1 + s2`

    D. String is a primitive type in Java

14. ____ **Which of the following defines a new type in Java? (All are valid Java statements.)**

    A. `Fruit fruit = new Fruit();`

    B. `#define Fruit {APPLE, ORANGE, BANANA}`

    C. `final boolean fruit = true;`

    D. `class Fruit{public bool citrus;};`

15. ____ **A Java error may be reported using**

    A. `assert width==15 : "Width isn't 15";`

    B. `System.err.println("Error! Width isn't 15");`

    C. `System.exit(-15);`

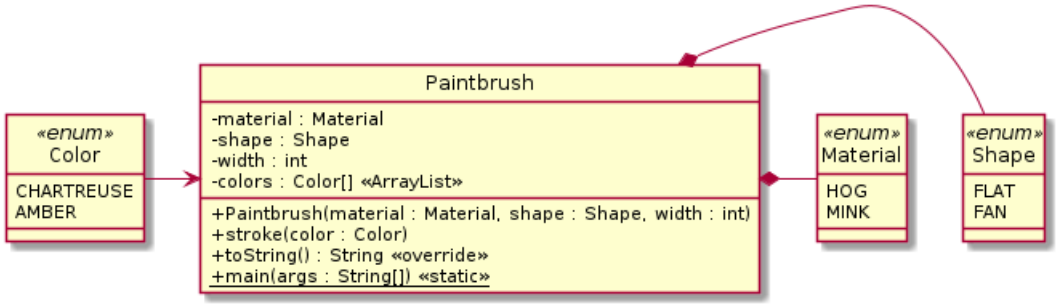
    D. All of the above

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. {Code a Class & Enum} Consider the class diagram below. Class `Paintbrush` represents an artist's brush constructed of hair from animals listed in `Material` cut in a particular `Shape` that is `width` wide (in millimeters). It also includes an `ArrayList` to store the `colors` painted. Paintbrush includes the usual constructor and `toString` method, a `stroke` method, and a static `main` method. **You will write only a small portion of this program.**



a. {3 points} In file Material .java, **write the Material enum type** with ONLY the values shown in the class diagram such that Material will be visible from any package in the system. Write ONLY the Material enum here: No other constructors, methods, fields, or classes are required. This is often a single line of code.

b. {5 points} In file Paintbrush.java, **begin writing class Paintbrush** shown in the class diagram. For this question, declare the Paintbrush class so that it will be visible to any code in the system, and then define all FOUR fields. DO NOT write the constructor and method members of class Paintbrush here - they may be written for separate questions below. Assume any required imports - you need not write them. Assume all enums are declared in the same package. Note that field `colors` is an ArrayList, NOT a simple array.

c. {5 points} In file Paintbrush.java, assuming all fields are properly declared and all enums are defined in the same package, **write just the Paintbrush constructor**, properly initializing all fields. Note that field `colors` is an ArrayList, NOT a simple array - it may be constructed inline or in the constructor.

In the consructor, also **perform data validation:** If parameter `width` is zero or less, throw an IllegalArgumentException with an appropriate message. (While this *may* require a "throws" clause, you may omit it or include it as you please.)

d. {4 points} In file Paintbrush.java, assuming all fields are properly declared and all enums are defined in the same package, **write the stroke method** that simply appends its parameter to field `colors`.

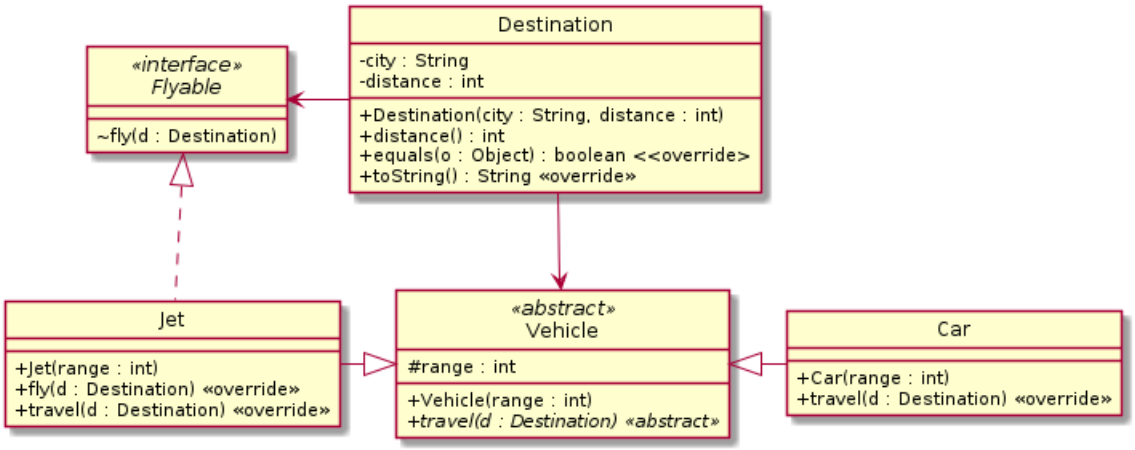
e. {5 points} In file Paintbrush.java, assuming all fields are properly declared and all enums are defined in the same package, **override the toString method** to convert all fields to a String as shown below. Use a for-each loop to include the `colors` ArrayList values. For example (do NOT return this as a *literal* String, but include the field values instead), a 20 mm paintbrush made of HOG hair in the FLAT shape that has painted strokes in colors CHARTREUSE and AMBER should return

```
HOG FLAT 20 mm paintbrush
   Painted CHARTREUSE
   Painted AMBER
```

f. {9 points} In file Paintbrush.java, assuming all fields are properly declared and all enums are defined in the same package, write static method `main`. In `main`:

- Instance ONE `Paintbrush` object named p of any valid width and shape and using any valid material.
- If an IllegalArgumentException or Exception (your choice) is thrown, catch it, print it to System.err, and then exit the program with a -1 error code.
- If no exception is thrown, use p to paint one `stroke` of any valid color.
- Finally, print p to the console.

2. {Inheritance} Consider the following class diagram. Class `Vehicle` represents an abstract machine for moving things from here to a `Destination`, with a maximum `range` it can travel without refueling and an abstract `travel` method to get to the `Destination`. Two subclasses of `Vehicle` are shown, a `Car` that represents ground transportation and a `Jet` that represents air travel and thus implements interface `Flyable`. You will only write small sections of this application!



a. {2 points} In file Flyable.java, write interface `Flyable`, which has a single required method `fly`. This is 2 or 3 short lines of code.

b. {2 point} If interface `Flyable` could have instead been written as an abstract class without breaking `Jet`, write it as an abstract class here. If not, write one *brief* sentence explaining why not.

c. {6 points} Write class `Jet`, which inherits from `Vehicle` and implements interface `Flyable`. `Jet` has 3 members: A constructor that chains to the `Vehicle` constructor, method `travel` (from `Vehicle`) that chains to the `Jet` method `fly`, and method `fly` that prints "Flying to " and its parameter to the console.

d. {6 points} For class `Destination`, implement ONLY method `equals` using the 4 steps discussed in the lecture. Two distinct `Destination` objects are equal if their `city` fields are equal, regardless of the `distance` field.

3. {3 points} Write custom exception `UtaException` with only the standard four constructors.

BONUS #1: {+2 points} Using ONE brief sentence, explain the difference between the association and the association class relationships.

BONUS #2: {+3 points} In no more than TWO brief sentences, explain what a destructor is AND why Java doesn't need them.