Full Name: _____

Student ID#: _____

CSE 1325 OBJECT-ORIENTED PROGRAMMING

PRACTICE #2 Exam #1 «---» 1 1 001 «---» Exam #1 PRACTICE #2

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	A block of memory associated with a symbolic name that contains a primitive data value or the address of an object instance
2	A template encapsulating data and code that manipulates it
3	An object created to represent an error or other unusual occurrence and then propagated via special mechanisms until caught by special handling code
4	A declaration that also fully specifies the entity declared
5	A class that cannot be instantiated
6	A procedure for solving a specific problem, expressed in terms of an ordered set of actions to execute
7	A program that runs in managed memory systems to free unreferenced memory
8	A special class member that creates and initializes an object from the class
9	Reuse and extension of fields and method implementations from another class
10	A style of programming focused on the use of classes and class hierarchies

Word List

Abstract Class	Abstract Method	Abstraction	Algorithm	Assertion
Class	Constructor	Data Validation	Declaration	Definition
Destructor	Encapsulation	Enumerated type	Exception	Field
Garbage Collector	Inheritance	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. {15 at 2 points each}

1. ____ **The two output streams that print to the terminal by default are**

   A. `System.out` and `System.err`

   B. `OUTPUT` and `CONSOLE`

   C. `System.term` and `System.log`

   D. `System.console` and `System.error`

2. ____ **An exception can only be caught if it is thrown**

   A. within the scope of a try

   B. by an assert statement

   C. within the scope of a catch

   D. by a library function

3. ____ **Subclass Student inherits from superclass Person. Method Student.age() can access field Person.birthday if birthday is**

   A. Protected or Public

   B. Public, Protected, or Private

   C. Protected or Private

   D. Student.age() can never access the fields of any other class

4. ____ **Field `int number;` is declared as protected. This means**

   A. number can be read as if public, but only written as if private

   B. number is only accessible from immediate class members

   C. number is accessible to all classes but not the main method

   D. number is accessible by members of this class and its subclasses only

5. ____ **Which Java constructor properly initializes field food?**

   A. `public Animal(Food food) : food = food { }`

   B. `public Animal(Food food) {this.food = food;}`

   C. `public Animal(Food food) {food = food;}`

   D. `public Animal(Food food) : food{food} { }`

6. ____ **In Java, we report errors using**

    A. `assert width==15;`

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

    C. `throw new Exception("Width isn't 15");`

    D. All of the above

7. ____ **Class Line includes public method x(). If x() is called using Line.x(), what is x()?**

    A. x() is a static method

    B. x() is an abstract method

    C. x() is a final method

    D. All of these must be true

8. ____ **The field** `final Fruit f;`

    A. must be assigned a value in the constructor, after which it may be changed

    B. will be assigned the value of its default constructor, since it is not initialized

    C. will result in a compiler error

    D. must be assigned a value once in the constructor, after which it may NOT be changed

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

    A. `class Bat extends Mammal, Flyer`

    B. `class Bat(Mammal, Flyer)`

    C. `class Bat implements Mammal && Flyer`

    D. Java doesn't support multiple inheritance for classes

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

    A. A closed diamond

    B. A closed arrow

    C. An open diamond

    D. An open arrow

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

    A. The copyright notice

    B. The name of the class and an optional stereotype

    C. The executable members such as constructors and methods

    D. The bottom pane is always empty

12. ____ **Which Java expression will append 42 to** `ArrayList<int> al`**?**

    A. `al++{42};`

    B. `al[al.size()+1] = 42;`

    C. `al.add(42);`

    D. `al += 42;`

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

    A. An int i may be converted to a String using casting, e.g., `(String) i`

    B. Each char in String s may be accessed using a for-each loop, e.g., `for(char c : s.toCharArray())`

    C. String is a primitive type

    D. In Java, String is another name for char*

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

    A. `` – ``

    B. `` ~ ``

    C. `` + ``

    D. `` # ``

15. ____ **Variables v1 and v2 are of type Vehicle. The expression** `v1 == v2`

    A. returns true if v1 and v2 reference exactly the same object

    B. returns the result of v1.equals(v2)

    C. returns true if the fields of v1 and v2 are equal

    D. generates a compiler error

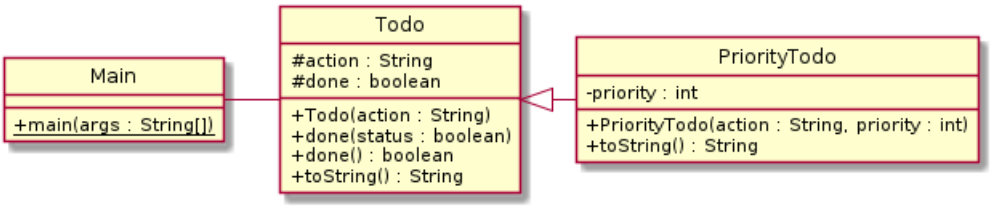
Free Response

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.**

1. {3 points} Circle each of the following that WOULD be inherited from a superclass in Java.

Constructor	Public Method	Package-private Method
Destructor	Protected Method	Private Method

2. Consider the following class diagram. Class Todo is a non-prioritized item on a "to do" list - that is, one task on a list that needs to be done. Class PriorityTodo is similar but also includes a priority number.

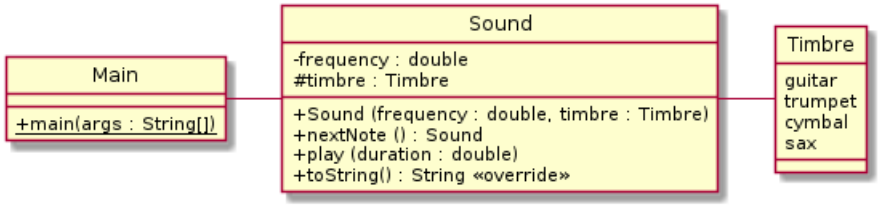


a. {5 points} Given the above UML class diagram, write T (for true) or F (for false) in the space before each statement that best describes it based on a theoretical Java implementation of the class diagram above. If asked to explain your answer, clearly state why you chose the answer you did **in one sentence**.

1. ____ Given `ArrayList<ToDo> todos = new ArrayList<>();`, the statement `todos.add(new PriorityTodo("Grade exam", 1))` will compile.
2. ____ PriorityTodo.toString cannot access Todo.done because Todo.done is not public.
3. ____ `Todo t = new Todo();` will compile, because Java always provides a default constructor.
4. ____ Class Todo will compile *even though* it has two methods named "done".
5. ____ PriorityTodo.toString() *should* be marked `@Override` in Java to override Todo.toString(), but this is not *required*.

b. {4 points} Write file PriorityTodo.java. In the constructor, initialize PriorityTodo.priority and use chaining to initialize Todo.action and Todo.done. Method toString() should return exactly what Todo.toString() returns concatenated with " priority " and PriorityTodo.priority.

Questions 3a through 3f are based on the following diagram, however, *each question is independent* of the others. Don't ignore the remaining questions if you have trouble answering one, just skip to the next question and continue.



Class Sound represents a musical sound, with a frequency (how "high" or "low" the tone is, from 20 to 20000 Hertz) and a timbre (what instrument the tone "sounds like", such as a guitar or a cymbal).

a. {4 points} In file Timbre.java, **write the Timbre enum type** with the four timbres shown in the class diagram such that Timbre will be visible from any package in the system. No constructors, methods, or fields are required.

b. {4 points} In file Sound.java, **begin writing class Sound** shown in the class diagram. Assume type Timbre is declared in file Timbre.java. For this question, write any imports required by your code, declare the Sound class, and then define all fields. Constructor and method members of class Sound may be written for separate questions below - do NOT write them here.

c {5 points} **As part of file Sound.java,** write just the constructor for class Sound, properly initializing all fields. Also perform data validation on the constructor field frequency - if outside of the specified range of 20 to 20000 Hertz, throw a IllegalArgumentException with the message "invalid frequency".

d. {4 points} **As part of file Sound.java,** write the nextNote method. Assume all required files are already included. The timbre of the new Sound is unchanged. The frequency will be the current object's frequency multiplied by 1.0595. Return the newly constructed object.

e. {4 points} **As part of file Sound.java,** overload the toString method such that if no superclass declares a valid toString, the compiler will generate the following: "error: method does not override or implement a method from a supertype". Otherwise, Sound.toString() should print the timbre, " @ ", the frequency, and " Hz" - that is, `Sound(Timbre.guitar, 400)` would print `guitar @ 400 Hz`.

4. Consider a simple library application.

a. {5 points} Write a custom exception named InvalidPageException, which inherits from RuntimeException. InvalidPageException has exactly one constructor, which accepts the book title (a String) and the requested page number (an int). The exception message should include the title and requested page number parameters. For example, if the book title is "Three Little Pigs" and the requested page number is 30, the exception message should be "Page 30 does not exist in 'Three Little Pigs'".

b. {4 points} For class Book below, write method selectPage. If parameter page is less than or equal to pages, set currentPage to parameter page. Otherwise, throw an InvalidPageException with the title and requested page number as parameters.

```
class Book {
    void selectPage(int page) {









    }
    String title = "Three Little Pigs";
    int pages = 12;
    int currentPage;
}
```

c. {5 points} For class Book below, complete static method main. Attempt to select page 30 in object book. If an InvalidPageException is thrown, print the exception message to the standard error channel.

```
public class Library {
    public static void main(String[] args) {
        Book book = new Book();









    }
}
```

BONUS {+3 points}

List the 3 types of intellectual property of most interest to software professionals with ONE sentence of description each. Do NOT write more than one sentence each!