CSE 230 Problem Set 07

Problem 23.1: Year Class

Consider the following class diagram:

Year
- y: Integer

+ display()
+ add(num : Integer)

Consider the following method definitions:

Pseudocode

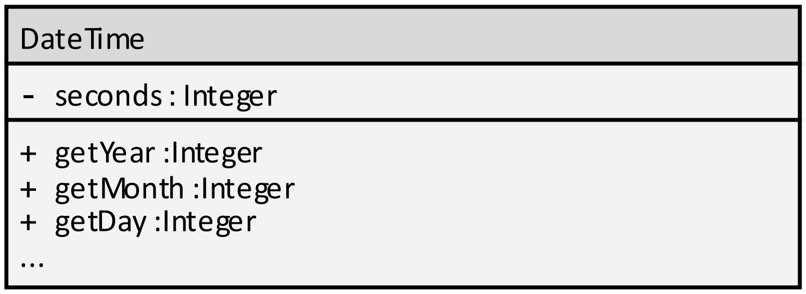
```
Year:: add(num)
   y += num

Year :: display()
   IF year > 0
      PUT year AD
   ELSE
      PUT -year BC
```

Classify the level of abstraction of the following class which stores a year as an int. Justify your answer. Hint: What happens when you subtract 2030 from today's year?

This class would be considered porous, a user needs to understand how to substract/input negative numbers in order to get proper BC years to work properly with this class.

Problem 23.2: Date-Time Class

Consider the following class diagram:



The Unix operating system represents time using the POSIX format. Here, time starts on the 1st of January 1970. Time is stored as a 32-bit integer, representing the number of seconds since that date. Classify the level of abstraction of the following class implementing POSIX date/time:

Pseudocode

```
DateTime::getYear
   RETURN seconds / 31,577,600
```

Classify the level of abstraction of the following class and justify your answer. Hint: What happens on the 19th of January 2038?

This class displays a critical level of abstraction. This class requires a full knowledge of the class and its implementation in order to properly use this class. Certain cases have to be specially handled in order for this class to not break.

Problem 23.3: File Name Class

Consider the following class diagram:

```
FileName
---------------------------------
-  name: String
-  extension: String
-  type: FileType
---------------------------------
…
+  isGreater(rhs: FileName) : Boolean
…
```

Consider the following method definitions: Note that the `isGreater()` method is used to sort files by their name so they are presented to the user in alphabetical order.

Pseudocode

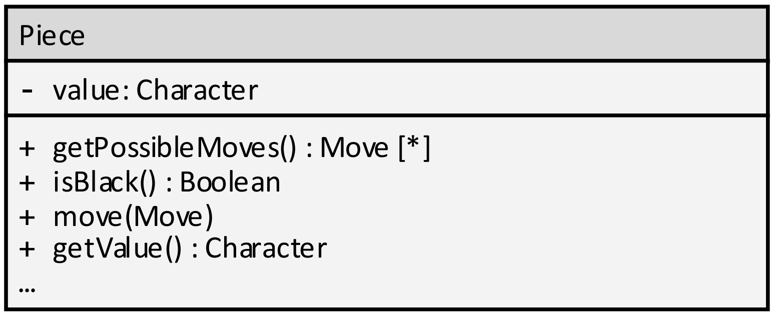
```
FileName :: isGreater(rhs)
   RETURN name > rhs.name
```

Classify the level of abstraction of the following class and justify your answer. Hint: What happens when I try to list “`a.txt`” and “`B.txt`” in the same directory?

This class can be classified as having porous abstraction. This class doesn’t require a detailed knowledge or understanding of its workings to use it, but users do need to know to use the same case styling(uppercase vs lowercase) for naming files in order for the isGreater method to work properly.

Problem 23.4: Chess Piece Class

Consider the following class diagram:



Classify the level of abstraction of a class designed to store a chess piece on a chess board. The member variable is a single character where ‘r’ corresponds to a white rook and ‘R’ corresponds to a black one. Note that the `getValue()` method returns the character corresponding to each chess piece.

This class can be classified as opaque. This is due to the getValue method simply returning the character corresponding to each chess piece instead of returning a string with the piece’s name. This character reveals a bit of the nature of the class.

Problem 23.5: Angle Class

Classify the level of abstraction of a class that stores an angle. This allows the client to work equally with radians (where 2π is a complete loop around a circle) and degrees (where 360° is a complete loop).

This class could be complete. Though there's not enough info to make a full evaluation, as long as there are enough public methods to fully understand how to use this class, it has complete abstraction. Otherwise, this class could be porous, in that it requires a bit of knowledge about the class in order to properly use it.