FINAL PRACTICE 3b **NAME:** _____

CS 207, Programming II

Computer Science - Fall, 2016

Problem 1:

Write the following program: CourseExamProcessing1.

You are given a data file named "course_exam_scores.txt", consisting of a list of exam scores, as integers of an undetermined size, one score per line. You do not know the number of scores in the list.

Your program is to do the following:

1. Read the scores into your program;
2. Display the scores, all values on a single line separated by a tab;
3. Calculate, as an integer value, the average of the scores;
4. Display the following message:
    a. "The average is *the calculated average*".
5. Handle the File I/O exception.

Below is a sample data set for the data file and the output generated from that data.

Note: your program is to be able to manage integer values of any size.

Sample course_exam_scores.txt file

```
87
68
59
44
99
92
88
15
97
83
93
```

Output generated for the above data file

```
/*PROGRAM OUTPUT

 The scores are the following:

 87 68 59 44 99 92 88 15 97 83 93

 The sum is  825
 The average is  75

*/
```

Problem 2:

Write the following interface and classes: IMachine, Machine, Calculator, Printer

The specification for each appears on the following pages.

Interface: IMachine

Write the interface: IMachine.

IMachine is to be defined as public interface.

The specification for IMachine is the following:

1) Three abstract, public methods:
    a. turnOn( ) which takes no parameters and has a return type of void;
    b. turnOff( ) which takes no parameters and has a return type of void;
    c. isOn( ) which takes no parameters and has a return type of boolean;

Class: Machine
Write the class: Machine.

Machine is to be defined as public, abstract class.

The specification for Machine is the following:

1) Data Members
    a. A data member `isOn` of type boolean with an access of private;
    b. A static data member `mCount` of type int. It is to have the access of public and it is to be initialized to zero.
2) Constructors
    a. A no-arg constructor that sets the data member `isOn` to false and increments `mCount` by one;
    b. A second constructor that receives a parameter of type boolean named isOn. The method is to do the following:
        i. Set the data member isOn to the value of the parameter;
        ii. Increment mCount by one;
        iii. Display the following message "A new machine has been created";
        iv. Increment mCount by one (note: this is in addition to step ii above);
3) Get/Set Methods
    a. A method setIsOn( ) that takes a boolean variable as a parameter and sets the data member isOn to the parameter value;
    b. A method getIsOn( ) that takes no parameters and returns the current value of the data member isOn.
4) Auxiliary Methods
    a. An abstract method displayIsOn( ) that
        i. takes no parameters;
        ii. has a return type of void;
        iii. has an access of type public;
    b. Override the IMachine abstract method turnOff( ) to
        i. display the message "Machine turned off!"
    c. Override the IMachine abstract method turnOn( ) to
        i. display the message "Machine turned on!"
    d. Override the IMachine abstract method isOn( ) to
        i. return true.

Class: Calculator
Write the class: Calculator.

Calculator is to be defined as a public class which is a subclass of Machine.

The specification for Machine is the following:

1) Data Members
    a. A static data member `calculatorCount` of type int. It is to have the access of private and is to be initialized to zero.
2) Constructors
    a. A no-arg constructor that increments the data member `calculatorCount` by one;
    b. A second constructor that receives a parameter of type boolean named isOn. The method is to do the following:
        i. Set the data member isOn to the value of the parameter;
        ii. Increment calculatorCount by one;
3) Auxiliary Methods
    a. A static method getCalculatorCount( ) that takes no parameters, has a return type of int and returns the current value of the data member calculatorCount;
4) IMachine method overrides
    a. turnOn( )
        i. sets isOn to true;
        ii. calls the parent class method turnOn( );
    b. turnOff( )
        i. sets isOn to false;
        ii. calls the parent class turnOff( );
    c. isOn( )
        i. returns the value of the parent method isOn( );
5) Machine method override
    a. Override the Machine class abstract method displayIsOn( ) to do the following:
        i. If the current state of isOn is true, the method displays the message “The current state of the calculator is on”;
        ii. If the current state of isOn is false, the method displays the message “The current state of the calculator is off”;
6) Object toString( ) override
    a. Override the Object toString( ) method to display the following:
        i. If the current state of isOn is true, display the message “The calculator is on and all is functioning well”;
        ii. If the current state of isOn is false, display the message “The calculator is off and it is not functioning at the moment”;

Class: Printer
Write the following class: Printer.

Class Printer is to be written as a subclass of class Machine.

Class Printer is to have the following two data members

1. a String named modelNumber that has an access of private;
2. a public static integer named pCount, of type int, that is initialized to zero and is incremented each time a new Printer object is created;

Class Printer will have a single constructor that takes two arguments, the first a boolean variable named isOn and a second a String variable named modelNumber that will set the corresponding class data members to those values. This constructor will do the following:

- set the appropriate data members;
- display the message "A new printer has been created";
- increment pCount by 1.

Class Printer will then override the IMachine abstract methods turnOn( ), turnOff( ), and isOn( ) by doing the following:

1. turnOn( ) will:
    a. display the message : "Warming up printer"
2. turnOff( ) will:
    a. display the message "Shutting down printer"
3. isOn( ) will;
    a. display the message "Printer is on" is the state of isOn is true;
    b. display the message "Printer is off" is the state of isOn is false;

Class Printer does not use the abstract method displayCount( ).

Class Printer will then add the method print which will have an access of public. The method print will:

1. receive a parameter of type int named "copies" that represents the number of copies to be printed;
2. display the message "The printer is printing". This message will be displayed the number of times that is represented by the value stored in the parameter copies. Each display is to appear on a separate line.

Finally, class Printer will override the Object class toString method. The Printer class toString( ) method is to display the following two messages:

"the machine is functioning properly"
"the printer is functioning properly".