

PURDUE UNIVERSITY

Midterm 2 Practice Examination– March 21, 2017

Problem-Solving and Object-Oriented Programming

Writing Time: TWO Hours

Question 1 (File Manipulation)

Phishing attacks using emails are still a large problem for many. A fast way for scammers to send out a large amount of “personalized” emails — making one more likely to fall for the scam — is to use file manipulation. Inside the **EmailScam** class, you will complete the method

```
public static void personalize(String name)
```

This method will read the **scam.txt** file, replace each hash symbol (#) with the name passed to the method, and write the personalized scam text to a file called **personalScam.txt** (Both text files are provided in the skeleton code folder).

Special cases:

- Return **false** if **null** is passed to the method, and do not edit the file
- Return **false** if an empty **String** is passed to the method, and do not edit the file

Question 2 (Inheritance)

1. The **Device** class, *which is provided for you*, implements the **Sellable** interface. Provide declarations for the methods **double getWholesalePrice()** and **double getRetailPrice()** in **Sellable**.
2. The **NotEnoughStorageException** is a subclass of **Exception**. Define this class, and provide a constructor **public NotEnoughStorageException(String message)** that calls the superclass's constructor, and passes the value stored in **message** to it.
3. The **Smartphone** class — a subclass of **Device** — is a representation of a smartphone, and contains two additional **private** fields — **numApps** and **storageCapacity**.

Within the **Smartphone** class:

- Provide a constructor **public Smartphone(double wholesalePrice, double retailPrice, int numApps, int storageCapacity)** to initialize your fields (instance variables)
- Provide accessors (getters) for your fields. Name them **getNumApps()** and **getStorageCapacity()**.
- Provide a method **public void downloadNewApp() throws NotEnoughStorageException** that increments the **numApps** field if the app count is less than the value of **storageCapacity** (just pretend that one app equates to one gigabyte of storage). Otherwise, the method should throw a **NotEnoughStorageException**, which you defined above.

Question 3 (Concurrency)

The class **SumArray** is a subclass of **Thread**. It is used to calculate the sum of an array using multiple threads, and contains two **private** fields — **nums** (of type **int[]**) and **sum** (of type **int**). **sum** is also **static**.

Within the **SumArray** class:

- Provide a constructor **public SumArray(int[] nums)** to initialize your array field (instance variable). **sum** should be initialized to zero *outside of the constructor*.
- Provide a **static** accessor (getter) method for the **sum** field named **getSum()**.
- Provide a method **public static void resetSum()** that resets **sum** to zero.
- Provide a method **public static synchronized void addToSum(int amount)** that adds to the **sum** field by the specified amount
- Override the method **public void run()**. It should calculate the sum of the elements in **nums** by adding each element value to **sum** (note: the use of **addToSum()** is *highly* recommended in your implementation of **run()**)

———— *End of Examination* ————