

OCTOBER 31, 2023

MIDTERM

EECS 3311

**YORK UNIVERSITY**  
**JULY 13, 2023 MIDTERM**  
EECS 3311  
SOFTWARE DESIGN  
Duration - 90 min  
Aids: None

*York University and you, as a student, share a commitment to academic integrity. You are reminded that you may be charged with an academic offence for using any unauthorized aids during the writing of an exam. A typical penalty for an academic offence may cause you to fail the course.*

*Please note: you **CANNOT** petition to **re-write** an examination once the exam has begun.*

*Good Luck!*

**Question 1.** [4 MARKS]

In each of the scenarios given below, name the design pattern that would be the most appropriate to implement.

- (a) You are implementing the data type Set as follows: for small sets, you want to use an array, but for larger sets, you want to use a hashtable. A Set object may start as an array, and once a threshold passes, it gets converted to a hashtable, and vice versa, once enough elements are deleted, it is reverted back to an array and so on.

---

- (b) You are implementing an arithmetic expressions evaluator where an arithmetic expression is parsed into a tree by the compiler.

---

- (c) You are working on a mathematical application that uses objects of a Matrix superclass. A Matrix object may be created using constructors of various Matrix subclasses, however client classes use same methods to manipulate them - such as add two matrixes, multiply two matrices, etc.

---

- (d) You are writing a gaming application. You need to implement a pattern that allows the player interface to be automatically notified whenever a friend of a player posts an update on the gaming discussion board.

---

**Question 2.** [2 MARKS]

Consider the following code:

```
class SuperSingleton implements Cloneable {

    @Override
    protected Object clone()
        throws CloneNotSupportedException
    {
        return super.clone();
    }
}

// Singleton class
class Singleton extends SuperSingleton {
    // public instance initialized when loading the class
    public static Singleton instance = new Singleton();

    private Singleton()
    {
        // private constructor
    }
}

public class Client {
    public static void main(String[] args)
        throws CloneNotSupportedException
    {
        Singleton s1 = Singleton.instance;
        Singleton s2 = (Singleton)s1.clone();
        System.out.println("s1 hashCode:- "
                           + s1.hashCode());
        System.out.println("s2 hashCode:- "
                           + s2.hashCode());
    }
}
```

Here is the output:

```
s1 hashCode:- 746299446
s2 hashCode:- 989310044
```

Clearly, we have two different instances of singleton. Please refactor the Singleton class so duplicated instances of Singleton are not allowed. You are not allowed to change neither SuperSingleton, nor the client class.

OCTOBER 31, 2023

MIDTERM

EECS 3311

*[Use the space below to write your solutions. ]*

**Question 3.** [6 MARKS]

Say you have a Fruit class with a pair of subclasses: Apple and Pear. You want to extend this class hierarchy to incorporate colors, so you plan to create Red, Green and Yellow subclasses. However, since you already have two subclasses, you'll need to create six class combinations such as RedApple, YellowPear and so on.

Adding new Fruit types and colors to the hierarchy will grow it exponentially. For example, to add a Mango fruit you'd need to introduce three subclasses, one for each color. And after that, adding a new color would require creating more subclasses, one for each fruit type. The further we go, the worse it becomes.

Using OO design principles and design patterns we learnt so far, solve the problem of class explosion.

Specifically, complete the answers to following questions:

1. ([1 mark]) Write in full the OO design principle that needs to be applied to solve the problem.
2. ([1 mark]) Identify the most relevant design pattern that can be used to solve the problem.
3. ([4 marks]) Draw the class diagram that may be used as a design of your solution. For each class provide class name, and any relevant properties and methods that may be necessary to solve the problem. Clearly identify relationships between them, and if necessary, provide multiplicities of the identified relationships if applicable. You may draw the diagram by hand or use any online tools suitable to draw class diagrams.

OCTOBER 31, 2023

MIDTERM

EECS 3311

*[Use the space below to write your solutions. ]*

**Question 4.** [4 MARKS]

When writing Java code for the subclass of a class, the first line of the constructor must be the call to `super()`. Please explain which of the solid principles mandates this. (2 marks).

Now consider this code:

```
public class IntSet {
    private Vector elements;

    public IntSet() {
        element = new Vector();
    }
}

public class MaxInteSet {

    public MaxIntSet () {
        // no code written here
    }

}
```

Is the principle identified in previous part is still broken? Explain why. (2 marks)



OCTOBER 31, 2023

MIDTERM

EECS 3311

*[Use the space below to write your solutions. ]*

EECS 3311

MIDTERM

OCTOBER 31, 2023

*[Use the space below to write your solutions. ]*