

CS 3321 or INFO 3307/5307

Homework 2 – Design Principles

Assigned: September 16, 2019
Due: October 04, 2019 @ 2300

Questions

In lecture, we discussed a number of software principles, including Encapsulate What Varies, DRY, LSP, OCP, ISP, and SRP. Identify in each of the following cases which if any of these principles are violated and if so, (a) justify your reasoning as to why that principle applies in a sentence or two, and (b) outline how the code can be corrected (words is fine for the latter if it is clear, or you can give revised code).

Note, there may be more than one principle violated. In the case of more than one, explain each principle; but, don't identify a design principle without very good cause.

1. (10 Points) This class is a snippet from a console-based 2D board game implementation (think e.g. Chess or Checkers)

```
class Game {
    private Piece[][] board;
    public void move() {
        /* Read user input */
        Point source = readCoordinateFromInput();
        Point dest = readCoordinateFromInput();
        /* Does the source square have a piece? Is the destination empty? */
        if (board[source.x][source.y] == null || board[dest.x][dest.y] != null)
            throw new IllegalMoveException();
        /* ... More rules here .. */
        /* Print out the current state of the board */
        printBoard();
    }

    private Point readCoordinateFromInput() {
        /* Read input from the terminal and convert to a point */
    }
    private void printBoard() {
        /* Printing code here */
    }
}
```

2. (10 Points) Here is another board game example, this time just code for the board and there are both 2D and 3D boards.

```
public class Square {
    int value;
}

public class Board {

    private Square[][] squares;
    public void updateSquare(int x, int y, Square aSquare) {
        squares[x][y] = aSquare;
    }

    public Square getSquare(int x, int y) {
        return squares[x][y];
    }

    // ... more methods
}

public class ThreeDBoard extends Board {

    private Square[][][] squares;
    public void updateSquare(int x, int y, int z, Square aSquare) {
        squares[x][y][z] = aSquare;
    }

    public Square getSquare(int x, int y, int z) {
        return squares[x][y][z];
    }

    // ... more methods
}
```

3. (10 Points) This code defines an API for interacting with a batch computing cluster

```
public interface ClusterManager {
    public int getJobCount();
    public List<JobId> getActiveJobIds();
    public JobId startJob(Job job);
    public double getJobProgress(JobId id);
    public void waitForJobToFinish(JobId id);
    public void pauseJob(JobId id);
    public void resumeJob(JobId id);
    public List<MachineId> getMachineIds();
    public MachineId getMachineIdForJob(JobId id);
    public List<JobId> getJobsOnMachine(MachineId id);
    public double getClusterLoadPercentage();
    public DiagnosticReport runClusterDiagnostics();
    public void restartCluster();
}
```

4. (10 Points) This is a code fragment for a multi-sensor HVAC (heating/air conditioning) system display.

```
class HVACController {
    // ...
    void updateDisplay () {
        double tenInsideTemps = 0;
        double tenOutsideTemps = 0.0;
        // implemented in/out temp so far ,
        // still have in/out humidity and 2nd floor temps to go
        for(int i=0; i<10; i++){
            tenInsideTemps = tenInsideTemps + insideSensor.getTemp();
            tenOutsideTemps = tenOutsideTemps + outsideSensor.getTemp();
            TimeUnit.SECONDS.sleep(1); // ten readings to average
        }
        double aveInsideTemp = tenInsideTemps / 10;
        double aveOutsideTemp = tenOutsideTemps / 10;
        display.setInsideTemp(aveInsideTemp);
        display.setOutsideTemp(aveOutsideTemp);
    }
}
```

5. (10 Points) This is code for a simple customer accounts system.

```
public class Customer {
    private String firstName;
    private String lastName;
    private String custNumber;
    private double balance = 0;
    // ... getters and setters too
}

public class Transaction {
    private Date date;
    private String description;
    private double amount;
    // ... getters and setters too
}

public class Account {
    private String acctId;
    private Customer customer;
    private Transaction[] transactions;

    public void deposit(double amount) { // ...
    }

    public void withdraw(double amount) { // ...
    }

    public double getBal() { // ...
    }
}
```

```
}  
  
public String getAcctId() { // ...  
}  
  
public void printStatement() { // ... print out all transactions  
}  
}
```

Submission

Submit a PDF file to Moodle by the deadline noted above. Any other file format will not be accepted. Late assignments will be handled according to the late assignment policy.