Name: ____

Number: ____

Hand in your examination paper and booklets used.

Answer all questions on the exam paper and any used booklets.

There are 5 questions and 50 marks.

Class diagrams must include classes, methods, associations, navigability, and multiplicities.

Sequence diagrams must include any required objects, classes, activation boxes, and must clearly show messages.

1. (6 marks)

a) What does the acronym *GoF* stand for? ____________________________

b) What design pattern is most closely related to the principle "*Don't call us, we'll call you*."?

____

c) If there are two possible behaviours for when *getInstance*() is received by a Singleton class, what type of instantiation is being used: lazy or eager? ___________________

d) Explain the difference between the Adapter and Façade patterns.

e) Suppose you have three classes *Dec1, Dec2, Dec3* that are considered decorators for another class named *Holiday*. Assume that each of these classes implement a method named *decorate*(). Illustrate the class diagram if these four classes collaborate according to the Decorator pattern.

f) Describe how the Interpreter pattern is similar to the Composite pattern.

2. (4 marks) Adapt the following situation to the Strategy pattern. Suppose a system has a class named *Item*, and where an algorithm is used to validate an instance of Item. Each instance of Item has a field *id* and a field *itemType*. There are 3 validation algorithms: *UPC*, *ISBN*, *Luhn*; the validation algorithm used depends on the item's type. For example, if itemType is "MasterCard" then the Luhn algorithm would be used to validate a 16 digit MasterCard number stored in id. But if itemType is "Book" then the ISBN algorithm would be used to validate a 13-digit ISBN stored in id.

Suppose the algorithm's validation method, *isValid,* returns a *boolean* value. *isValid* has one parameter *id*. For instance, a method of Item could include the line:

```
if (algorithm.isValid(id)) System.out.println(id+" is valid");
```

a) Draw a class diagram.

b) Draw an object diagram for an item that is validated according to the Luhn algorithm.

3. (10 marks) Consider the following code

```
public class Driver
{
   public static void main
                  (String[] args)
   {
      Turnstile t = new Turnstile();
      t.coin();
      t.pass();
      t.pass();
      t.coin();
      t.coin();
   }
}
public class Turnstile {
   State state;
   public Turnstile() {
      state = new Locked(this);
   }
   public void coin() {
      state.coin();
   }
   public void pass() {
      state.pass();
   }
}
public interface State {
   public void coin();
   public void pass();
}
```

```
public class Locked  implements State {
   private Turnstile t;
   public Locked(Turnstile t) {
      this.t = t;
   }
   public void coin() {
      t.state = new Unlocked(t);
      System.out.println("thank you ... please proceed");
   }
   public void pass() {
      System.out.println("hey ... come back here");
   }
}
public class Unlocked implements State {
   private Turnstile t;
   public Unlocked(Turnstile t) {
      this.t = t;
   }
   public void coin() {
      System.out.println("just one coin ...");
   }
   public void pass() {
      t.state = new Locked(t);
      System.out.println("bye bye");
   }
}
```

a) based on the code, draw a class diagram.

b) based on the code, draw a statechart diagram.

d) show the output produced when *main()* executes.

c) based on the code, draw a sequence diagram to illustrate all messages sent when *main()* executes.

4. (10 marks) Consider a language used to compare text strings where we are able to determine if one string contains another string,
determine if one string is equal to another string,
and to combine expressions using And and Or logic.
We define the language as:

```
expression ←  containsExpression | equalsExpression | andExpression | orExpression

string ←letter | letter  string

letter ← 'a' | 'b' | 'c' | … | 'z'

containsExpression ←string IN string

equalsExpression ←string EQUALS string

andExpression ←expression AND expression

orExpression ←expression OR expression
```

To determine if a string, say "university of winnipeg", contains "it" and either "peg" or "win" we can construct a sentence in the above language and evaluate its corresponding syntax tree.

a) What is the required sentence expressed in the language?

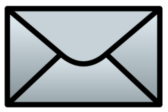
b) What is the syntax tree for the sentence in a)?

c) Illustrate the class diagram that results when we apply the Interpreter design pattern to the above language.

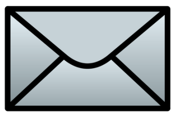
5. (10 marks) Suppose we are designing a game based on the problem we shall refer to as *The Two Envelopes Problem*.

The game has two envelopes; each envelope contains money. One envelope contains twice the amount of money of the other. Initially each envelope is Green in colour.

The game begins with a player asked to make a choice: Pick one of two envelopes. Once the player picks an envelope, say envelope1, then instead of immediately opening that envelope, the game asks the player, "Do you want to change your choice and pick the other envelope, envelope 2?" The player can then change their choice, but then the chosen envelope opens displaying its contents.

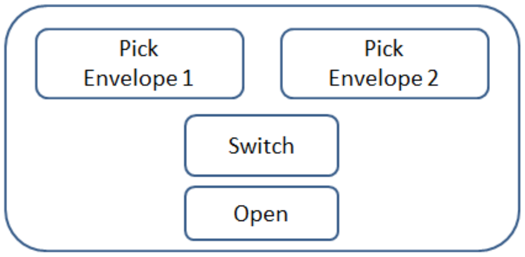


Envelope 1



Envelope 2

In our software version of this game there is a remote control with four buttons:



Each of the top 2 buttons relates to the player's initial choice. Pressing one of these results in the chosen envelope changing its colour to Red to show that it was chosen.

After pressing one of the two Pick buttons the game displays its message "Do you want to change your choice and pick the other envelope?" Now the player will either press Open, or, Switch.

The button, Switch, results in the previously chosen envelope changing its colour back to Green, the other envelope changing its colour to Red, and then the envelope opens.

a) Draw a class diagram that illustrates the use of the Command design pattern in our game's design. The class diagram may not be complete – you must show what is relevant to the Command pattern and our knowledge of the game. Note that an Envelope has methods such as *changeColour(String colour)* and *revealContents()*.

b) For the following assume:

- The player has made a first choice, Envelope1.
- The game has asked the player "Do you want to change your choice and pick Envelope 2?"

Use a sequence diagram to show the messages that you would plan to be sent if the player were to now press the Switch button.

6. A software auction can be designed based on the Observer pattern. Consider the following that describes how a normal non-computerized auction is handled:

There is one auctioneer and several bidders

The auctioneer controls the bidding process

Several bidders can register for the auction (and receive a numbered paddle)

Each bidder has a numbered paddle he/she uses to place a bid

The auctioneer informs all bidders that the auction i20s beginning

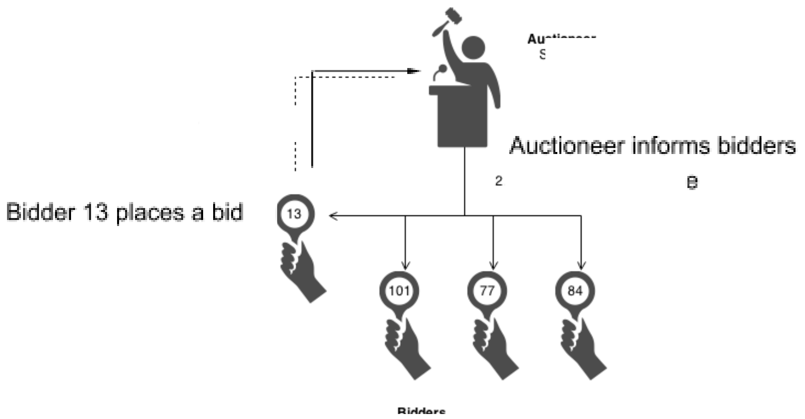
The auctioneer informs all bidders of the initial bid price

The auctioneer waits for a bidder to place a bid

When a bidder places a bid the auctioneer accepts the bid and raises the bid price

When the auctioneer changes the bid price then the new price is broadcast to all bidders

If no bidder places a bid within a specific period of time the auctioneer informs the bidders that the auction is closed (i.e. the auction has ended).



a) Design a software auction system based on the Observer pattern. Draw a class diagram and include classes such as Auctioneer and Bidder. Remember to include methods for the running of an entire auction from start to end.

b) Draw an object diagram to show an auction, an auctioneer and 3 bidders.

c) Draw a sequence diagram that shows
   - the auctioneer informing the bidders that the new bid price is $100, followed by
   - one of the bidders placing a bid at $200, followed by
   - the auctioneer informing the bidders that the auction is closed