Inclass test for SOEN 345, Prof Rigby

Worth 15% of total grade. Don't cheat, 15% isn't worth the black mark on your academic record.
Answer all questions in the exam booklet!

Multiple Choice [3 marks]

Question 1: Mockito mocks objects using:

A. Reflection and a proxy object
B. Reflection and a facade object
C. Reflection of an Adapter object
D. All of the above

Question 2: What is the output for the following code?

```
when(mockedStack.pop()).thenReturn(3,2,1);
mockStack.pop(); mockStack.pop(); mockStack.pop(); mockStack.pop();
```

The output is:

A. 3, 2, 1, 1
B. 3, 2, 1
C. 1, 2, 3
D. 3, 2, 1, null

Question 3: Git bisect on the following sequence of commits. You get fail, pass, fail, which is the culprit commit?

1----2----3----4----5----6----7----8----9

Short Answer [5 marks]

Question 4: What is Feathers's definition of "legacy code?" [1 mark]

"code without tests"

Question 5: What are the two key characteristics of a unit test? [2 marks]

"Test runs fast" & "Test help localize problems"

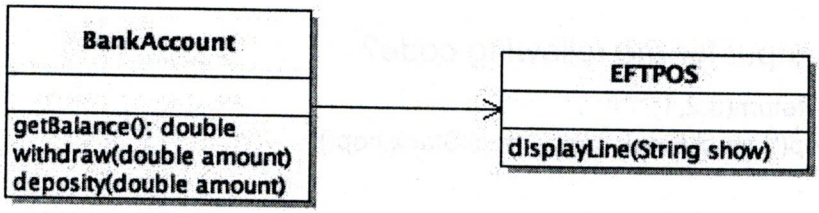
Question 6: Name the type and give an example of a flaky test? [2 marks]

See printed slide.

Coding Questions [14 marks]

Question 7: Recall that the *iterator* for the Stack class for Java is incorrect. Write a unit test that would characterize the actual behaviour of the Java Stack's iterator. [4 marks]

Question 8: The BankAccount code is dependent on the old EFTPOS terminal. [6 marks]



1. Draw a new diagram with the dependency on EFTPOS broken
2. Add a new iPad terminal to your diagram
3. Write Mockito code to check that the BankAccount calls the iPad class to display the balance when a withdrawal is made

Question 9: Parameterize constructor [4 marks].

Consider the following constructor:

```
Pixel(Color color) {
        this.color = color;
        this.position = new position(23, 4, 52);
}
```

1. Parameterize the constructor
2. Make the change behaviour preserving
3. Mock out the position and color

Question 6.

Sources of non-determinism

- Inherent non-determinism
  - Noisy or complex test environment introduces non-determinism
    - Sometimes can fix environment
  - Asynchronous calls introduces non-determinism
- Accidental non-determinism
  - Old and out-of-date tests introduce flakiness (ie accidental non-determinism)

Question # 7

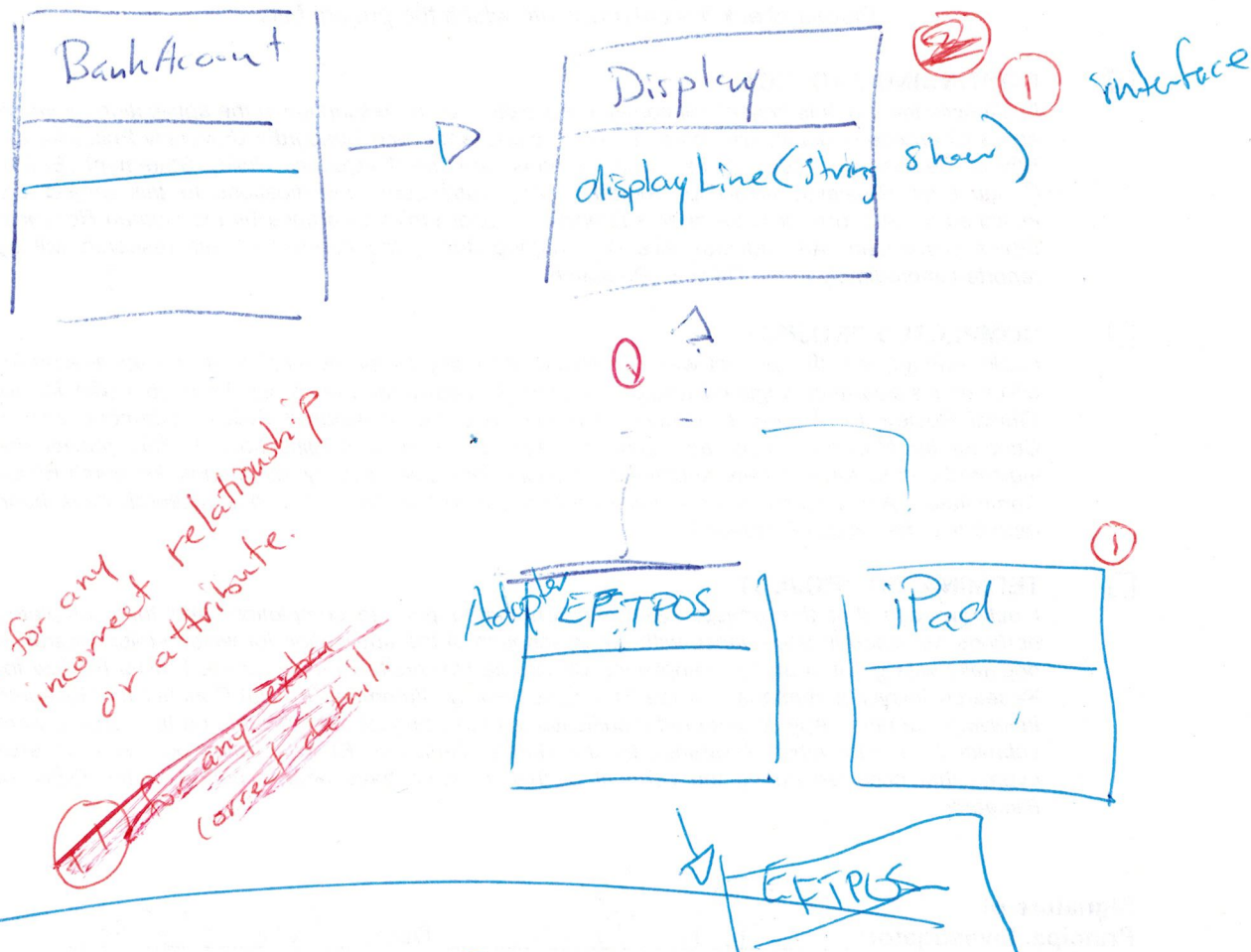
```
Stack<Integer> stack = new Stack<Integer>();

stack.push(1);
stack.push(2);

Integer val = 1;
for (Integer i : stack) {
    assertEqual(val, i);
    i++
}
assertFalse.isEmpty();
```

① push more than one element

② - iterator
- get the order of a queue

① shouldn't be empty.

Question 8



-1 for any incorrect relationship or attribute.

~~Any other correct detail.~~

```
// create mock of iPad
iPad ipad = new mock(iPad.class);
BankAccount ba = new BankAccount(ipad);
ba.withdraw(13.00);
verify(ipad).showLine(13.00);
```

-2 if mock BankAccount

Question #9

```
Pixel (Color color, Position position) {
    this.color = color;
    this.position = position;
}

Pixel (Color color) {
    this (color, new Position (23, 4, 52))
}

Test () {
    Color color = new mock (Color.class);
    Position position = mock (Position.class);
    Pixel pixel = new Pixel (color, position);
}
```

(1) (1) (1)

+1 for having two constructors & assigning vars

~~if one constructor~~

mock pixel (-2)

code understood