CSC 207H1 S 2013 Final exam
Duration — 3 hours
Aids allowed: 2-sided aid sheet
(max A4-sized)

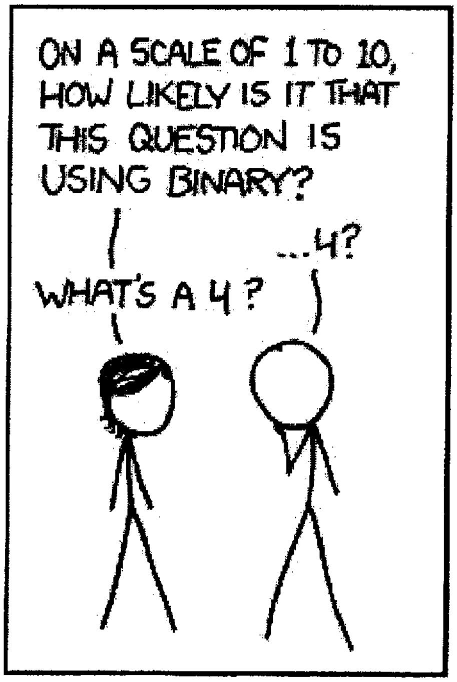
Student Number: └┴┴┴┴┴┴┴┴┴┘

Last Name: ____________________ **First Name:** ____________________

Lecture Section: L0101 Instructor: Jonathan Lung

Do **not** *turn this page until you have received the signal to start.*
Remember that a mark of 40% or higher on this exam is required to pass this course.
Please fill out the identification section above and read the instructions below.
Good Luck!

This exam consists of 5 questions on 16 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.* If you use any space for rough work, indicate clearly what you want marked.


ON A SCALE OF 1 TO 10, HOW LIKELY IS IT THAT THIS QUESTION IS USING BINARY?
...4?
WHAT'S A 4?

Source: http://xkcd.com/953/

# 1:	______/20
# 2:	______/20
# 3:	______/15
# 4:	______/20
# 5:	______/20
TOTAL:	______/95

Total Pages = 16

Question 1. [20 MARKS]

Short answer questions

Part (a) [1 MARK]

What design pattern(s) might you expect to see in an implementation of a stack in which the items contained need to be iterated through?

Part (b) [3 MARKS]

Suppose you have a Java program in a file named `StyleChecker.java` in an otherwise-empty directory. What command(s) would you run from the command line in order to start the program? Assume that the code works, that the `StyleChecker` class is not in a package, the program does not need a special class path set, and the program does not need arguments from the command line.

Part (c) [1 MARK]

What method(s) **must** be defined in `StyleChecker.java` for the above command(s) to succeed?

Part (d) [3 MARKS]

Suppose you have a field (instance variable) in a large project that needs to be renamed. Briefly describe three situations in which performing a regular find-and-replace operation on the `.java` that defines the field will produce incorrect results. You may use point form.

Part (e) [2 MARKS]

Give **exactly** four examples where documentation or other communication (in the broadest sense of the word) may be required while developing software as part of a team.

Part (f) [2 MARKS]

Consider the following regular expression:

`^[01]-[89]*$`

Label the four string below that contain a string that will be matched by this regular expression. Do this by circling any string that is matched by the regular expression and putting an "X" next to any string that will not be matched.

There is a penalty for guessing (0.5 marks off per incorrect answer). You can receive a negative mark for this question.

‘‘25’’	‘‘25-30’’
‘‘1-9’’	‘‘1-99’’

Part (g) [2 MARKS]

Consider the following regular expression:

`^(?:ABC|DEF?)([^I]+)G(.*?)H$`

For each of the rows in the table below, do the following:

In columns 2 and 3, write the contents of capturing groups 1 and 2, respectively, when the string in column 1 is matched.. Write "no match" in each table cell where a match is not found. One row has been filled out for you.

String	Group 1	Group 2
GH	No match	No match
ABCGH		
ABCDEFGH		
ABCDEFGH ABCDEFGH		
DEFEND GINGER UH		

Part (h) [2 MARKS]

A jPhone model name looks like `jPhone` x`GB` where x is an integer representing the storage capacity of the jPhone. The storage capacity of a jPhone is made up of one or more digits and may contain leading 0s.

Write a regular expression that contains exactly one capturing group that will capture all the characters representing the storage capacity of a jPhone, including any leading 0s. The capturing group shall not capture any extra characters.

Part (i) [2 MARKS]

An undergraduate course code such as CSC207 consists of a subject code (for example, CSC) followed by a course number (for example, 207). The course number always follows the subject code and may optionally be separated by a single space character. The subject code is three uppercase alphabetic characters (from the Roman alphabet) long and the course number is a three digit number. The first digit of the course number is either 1, 2, 3, or 4.

Graduate course codes are similar. The only difference is that the course number portion of the course code is exactly four digits long instead of three (for example, `CSC1234`) and the only other restriction on the number is that the first digit is not permitted to be 0.

Write a regular expression that will match strings that contain **only** strings that could be **undergraduate** course codes and that will match no other strings. For example, `ABCD123` should **not** be matched, even though it *contains* an undergraduate course code.

Part (j) [2 MARKS]

Continued from (i)

Write a regular expression that will match strings that contain **only** strings that could be undergraduate or graduate course codes and that will match no other strings. For example, your expression should match both `ABC1234` (a graduate course code) and `DEF 123` (an undergraduate course code).

Question 2. [20 MARKS]

Compare and contrast the items below (that is, describe the similarities and differences between them). You may use point form.

Part (a) [2 MARKS]

Returning an `Exception` (or a subclass of it) and *raising* an `Exception` (or a subclass of it).

Part (b) [3 MARKS]

Product backlog and sprint backlog.

Part (c) [3 MARKS]

The commands `svn update` the command `svn commit`.

Part (d) [3 MARKS]

An Java interface and a subclass.

Part (e) [3 MARKS]

The heap and the stack in the Java memory model.

Part (f) [3 MARKS]

A `.class` file and a `.java` file.

Part (g) [3 MARKS]

An Java instance variable declared `private` and a Java instance variable declared `protected`.

Question 3. [15 MARKS]

For each of the following sets of code, state the reason(s) why the file(s) will not compile. If the files *do* compile, state the reason(s), if any, that will cause one or more of the methods to raise an exception. Be sure to indicate if the error is caught at compile-time or at run-time. If no errors would be encountered, write "no errors". Note that some sets of code include more than one file listing. You may use point form.

Part (a) [2 MARKS]

Listing 1: ClassA.java

```
public class ClassA {
    public static int x = 0;

    public void setX(int x) {

    }
}
```

Answer:

Part (b) [2 MARKS]

Listing 2: ClassB.java

```
public class ClassB {
    public static int x = 0;

    public void setX(int y) {
        this.x = y;
    }
}
```

Answer:

Part (c) [2 MARKS]

Listing 3: ClassC.java

```
public class ClassC {
    public int x = 0;

    public void getX() {
        return this.x;
    }
}
```

Answer:

Part (d) [2 MARKS]

Listing 4: ClassD1.java

```
public abstract class ClassD1 {
    protected int x;

    public ClassD1() {
        this.x = 0;
    }

    public abstract void increment();
}
```

Listing 5: ClassD2.java

```
public class ClassD2 extends ClassD1 {
    public void increment(ClassD1 d1) {
        d1.x = d1.x + 1;
    }
}
```

Answer:

Part (e) [2 marks]

Listing 6: ClassE1.java

```
public abstract class ClassE1 {
    protected int x;

    public ClassE1() {
        this.x = 0;
    }

    public void increment() {

    }
}
```

Listing 7: ClassE2.java

```
public class ClassE2 extends ClassE1 {
    public void increment() {
        this.x = this.x + 1;
    }
}
```

Answer:

Part (f) [2 marks]

Listing 8: ClassF.java

```
public class ClassF {
    public Integer x;

    public void setX(Object a) {
        this.x = (Integer) a;
    }

    public void call() {
        this.setX("0");
    }
}
```

Answer:

Part (g) [3 MARKS]

Listing 9: ClassG1.java

```java
public class ClassG1 {
    public int x;

    public ClassG1() {
        this.x = 0;
    }
}
```

Listing 10: ClassG2.java

```java
public class ClassG2 {
    public int getX() {
        return x;
    }
}
```

Listing 11: ClassG3.java

```java
public class ClassG3 extends ClassG1, ClassG2 {
    public int getY() {
        return 0;
    }
}
```

Answer:

Question 4. [20 MARKS]

Read the following problem description and come up with a good program design.

The city of Computo is modernizing its library system and wants an electronic catalogue (list of items) and checkout system to allow visitors to borrow materials. The library contains various materials: books, magazines, CD-ROMs, and DVDs. Each of these has a list of details: a reference number, a (unique) barcode number, a title, and publication year. All users of the library system should be able to search for books by reference number or title and be able to see those pieces of information.

Visitors to the library may request a library card from the library staff (visitors may have more than one card); each card has a unique number. Library staff can then have a card issued by the system and give this card to the visitors. Library staff also need to be able to add newly purchased materials to the library catalogue.

By inserting a library card into a machine at the library, visitors can

- check out materials from the library that are not already checked out (checked out materials are borrowed from the library and must be returned after a fixed of time),
- renew (that is, double the borrowing period) any materials they currently have checked out; visitors may only renew a book once per checkout,
- view what materials they have checked out along with their due dates,
- pay fines for missing and overdue materials.

Fines for adults are $2 per day and fines for children are $1 per day. Fines for missing books are $200 for everyone. All other materials cost $100 to replace. The borrowing period for CD-ROMs and DVDs is one week. All other materials can be borrowed for two weeks.

Furthermore, the system should be able to print out a list of names and addresses of visitors that have unpaid fines. A visitor can return a item by inserting into a return slot.

Outline your design here by writing the Java classes, methods signatures, and field declarations (static and instance variables and constants). In particular, you do **not** need to fill in the body of any methods nor do you need to leave any comments (although you may use comments to clarify your answer). The next three pages have been left blank to provide space for your answer. Your program design does not need to accommodate saving to disk.

Possibly handy Java API notes

```
class Collections:
   static sort(List list) // sort list
interface Collection:
   add(Object e) // adds e to the Collection
   boolean isEmpty() // returns true iff this Collection is empty
interface Set extends Collection, Iterator:
   // A Collection that models a mathematical set; duplicates are ignored.
class HashSet implements Set
interface List extends Collection, Iterator:
   // A Collection that allows duplicate items; a List has order
class ArrayList implements List
interface Map extends Collection:  // An interface for objects that map keys to values.
   boolean containsKey(Object k) // returns true iff this Map has k as a key
   Object get(Object k) // returns the value associated with k or null if k is not a key
   Set keySet() // returns the set of keys
   put(Object k, Object v) // adds the mapping k -> v
class HashMap implements Map
```

Question 5. [20 MARKS]

Essay questions. If you leave an answer blank in this section or write "do not mark", you will receive 2 marks for the question and the rest of your answer will not be marked.

Part (a) [10 MARKS]

In this course, you were given seven weeks to complete an assignment with several other people using agile methods. Supposing the course had two four-week long programming assignments (with no product owner) to be done individually, which tools, techniques, and strategies from this course do you think you would you use and why? Which would you not use and why?

Part (b) [10 MARKS]

How do you think things would have been different if you had 10 team members for both assignments 1 and 2? Why?