

UNIVERSITY OF TORONTO  
Faculty of Arts and Science  
DECEMBER 2014 EXAMINATIONS  
CSC207H1F  
Duration — 3 hours  
No Aids Allowed

Student Number: \_\_\_\_\_

Last Name: \_\_\_\_\_ First Name: \_\_\_\_\_

Instructors: **Mohamed A. Mansour**  
**Michalis Famelis**  
**Emad Andrews**

Please fill out the identification section above as well as the  
one on the back page, and read the instructions below.

Good Luck!

This quiz consists of 10 questions on 26 pages (including this one). You must achieve 30% in this test to pass the course. When you receive the signal to start, please make sure that your copy of the exam is complete.

If you use any space for rough work or need to scratch out an answer, clearly identify the part that you want us to mark.

Question		Out Of	Mark
1.	Multiple Choice Questions #1	12	
2.	Multiple Choice Questions #2	8	
3.	Version Control	6	
4.	Polymorphism	12	
5.	Testing	10	
6.	Generics and Collections	10	
7.	UML	10	
8.	Exceptions	10	
9.	Design Patterns	10	
10.	Regular Expressions	12	
	TOTAL	100	

(This page is intentionally left blank.)

## Question 1 [12 marks]

For the following questions, **circle** the correct answer.

### Part (a) [1 mark]

The design pattern that could be paraphrased as follows: "Please notify me whenever you change yourself."

- (a) Observer
- (b) Singleton
- (c) MVC
- (d) Factory

### Part (b) [1 mark]

Suppose class Sneetch implements Observer. Which statement is true? Circle the true statement.

- (a) Sneetch inherits a method called update that it must not override
- (b) Sneetch inherits a method called update that it has the option of overriding
- (c) Sneetch must implement a method called update
- (d) Trick question! This is impossible because \_\_\_\_\_

### Part (c) [1 mark]

Suppose class Smurf implements Observable. Which statement is true? Circle the true statement.

- (a) Smurf must implement a method called NotifyObservers()
- (b) Smurf must not extend anything
- (c) Smurf must not implement anything else
- (d) Trick question! This is impossible because \_\_\_\_\_

### Part (d) [1 mark]

When class C1 follows the Singleton design pattern, which of the following must you ensure? Circle the true statement.

- (a) There can never be more than one instance of type C1
- (b) Each instance of C1 must have exactly one instance variable
- (c) Each instance of C1 must have at most one instance variable
- (d) There must be a single parent class C1

### Part (e) [1 mark]

A program involves Sneetch and Smurf classes, if I tell you that every Sneetch is a Smurf, which design makes the most sense?

- (a) Class Sneetch is a parent of class Smurf.
- (b) Class Smurf is a parent of class Sneetch.
- (c) Classes Sneetch and Smurf are both children of a common parent class.
- (d) A common child class has both class Sneetch and class Smurf as parents.

**Part (f) [1 mark]**

For any class that does not have any constructors

- (a) Java will define one
- (b) Eclipse will define one
- (c) The code will not compile until the developer explicitly creates one
- (d) The code will compile but it will throw a run time exception when it runs.

**Part (g) [1 mark]**

An abstract class can be instantiated as long as:

- (a) It has at least one method that is not abstract
- (b) All its methods are not abstract
- (c) None of the above since the abstract class can never be instantiated

## Question 1: Parts (h) to (l)

For the rest of the question, consider the following code, that compiles without errors:

```
public class A {
    int num = 13;
    public A() {
        System.out.print("one");
    }

    public void report() {
        System.out.print("two");
    }
}

public class B extends A {
    int num = 2;
    public B() {
        System.out.print("here");
    }
}

public class C extends B {
    public C() {
        System.out.print("some");
    }
    public void report() {
        System.out.print("all");
    }
}
```

Suppose we have a main method in another class that says: `A var1 = new C();`  
This compiles and runs without error.

### Part (h) [1 mark]

What output will be created by constructing that instance of C?

- (a) some here one
- (b) one here some
- (c) some

### Part (i) [1 mark]

If the expression `var1.num` is then used, which variable is accessed?

- (a) The instance variable num in class A
- (b) The instance variable num in class B
- (c) The instance variable num in class C
- (d) None of the above; this expression is illegal

**Part (j) [1 mark]**

If the expression ((C) var1).num is then used, which variable is accessed?

- (a) The instance variable num in class A
- (b) The instance variable num in class B
- (c) The instance variable num in class C
- (d) None of the above; this expression is illegal

**Part (k) [1 mark]**

If the method call var1.report() is then used, which method is called?

- (a) The method report in class A
- (b) The method report in class B
- (c) The method report in class C
- (d) None of the above; this method call is illegal

**Part (l) [1 mark]**

If the method call ((B) var1).report() is then used, which method is called?

- (a) The method report in class A
- (b) The method report in class B
- (c) The method report in class C
- (d) None of the above; this method call is illegal

**Question 2** [8 marks]

For each question below, a correct answer earns 2 marks, "I don't know" earns 0 marks, and an incorrect answer earns -1 mark. **Do not guess.**

**Part (a)** [2 marks]

Consider the following program, which does not have syntax errors.

```
public class MyProgram {
    public static void throwit() {
        throw new RuntimeException();
    }
    public static void main(String args[] ) {
        try {
            System.out.println("Hello");
            throwit();
            System.out.println("Done");
        } finally {
            System.out.println("Finally");
        }
    }
}
```

Which of the following most closely describes the behaviour of the program above? Circle one.

- (a) The program will not compile
- (b) Print Hello, then report a RuntimeException, then print Done, then print Finally
- (c) Print Hello, then report a RuntimeException, then print Finally
- (d) Print Hello, then print Finally, then report a RuntimeException
- (e) I don't know

**Part (b)** [2 marks]

Consider the following program:

```
public class MyProgram {

    public static void throwit() {
        throw new Exception();
    }

    public static void main(String args[] ) {
        try {
            System.out.println("Hello");
            throwit();
            System.out.println("Done");
        } finally {
            System.out.println("Finally");
        }
    }
}
```

```
    }  
}
```

Which of the following most closely describes the behaviour of the program above? Circle one.

- (a) The program will not compile.
- (b) Print Hello, then report an Exception, then print Done, then print Finally
- (c) Print Hello, then report an Exception, then print Finally
- (d) Print Hello, then print Finally, then report an Exception
- (e) I don't know

**Part (c) [2 marks]**

Consider this program:

```
public class TryIt {  
    public static void main (String [ ] args) {  
        Integer i = new Integer(42);  
        Integer j = new Integer(42);  
        System.out.println((i == j) + " " + i.equals(j));  
    }  
}
```

What is the output of the program above? Circle one.

- (a) true true
- (b) true false
- (c) false true
- (d) false false
- (e) I don't know



**Part (d) [2 marks]**

Consider this program:

```
public class Document {
    public static int quantity = 0;

    public Document() {
        quantity = quantity + 1;
    }

    public static int getQuantity() {
        return quantity;
    }
}

public class Book extends Document {
    private int numChapters;

    public Book(int numChapters) {
        this.numChapters = numChapters;
    }

    public static void main(String[ ] args) {
        Book b1 = new Book(4);
        Book b2 = new Book(6);
        Document d1 = new Document();
        System.out.println(Document.getQuantity());
    }
}
```

What is the output of the program above? Circle one.

- (a) 0
- (b) 1
- (c) 2
- (d) 3
- (e) I don't know.

### Question 3 [6 marks]

#### Part (a) [4 marks]

Fill in the blanks:

My team-mate emailed me that she had committed her changes to the design document in our Subversion repository. To see her changes, I \_\_\_\_\_ my local copy. I then used the command `svn _____` to read her latest commit message. I looked into the file she had changed. I noticed she had made a typo, so I fixed it. When I ran the command `svn _____`, I saw the following output:

```
M    design.txt
```

Finally, I committed my changes using the command:

```
svn commit -m "_____"
```

#### Part (b) [2 marks]

If you checkout the Subversion repository that Goran uses for his personal documents, you can find the following files:

- letter.txt
- letter-FINAL.txt
- letter-old.txt
- letter-updated-version1.txt
- letter-updated-version2.txt

Briefly explain what Goran is doing wrong.

**Question 4** [12 marks]

Assume that Java allowed multiple inheritance. Consider the following code:

```
public class Ko {  
    public void koko(){  
        System.out.println("koko");  
    }  
}
```

```
public class Lo extends Ko{  
    Public void koko(){  
        System.out.println("loko");  
    }  
}
```

```
public class Mo extends Ko, Lo {  
}
```

```
public class EstasLoco {  
    public static void main(String args[ ]){  
        Mo mimi = new Mo();  
    }  
}
```

**Part (a)** [1 mark]

Below, write a single line of code that uses the object mimi declared in `EstasLoco#main()` to expose the ambiguity introduced by multiple inheritance.

**Part (b)** [1 mark]

Briefly explain what the ambiguity is.

**Part (c)** [2 marks]

Java does not allow multiple inheritance. However, a class can implement multiple interfaces. Consider the following code:

```
public interface IKo {  
    public void ikoko();  
}
```

```
public interface ILo {  
    public void ikoko();  
}
```

Write a new class named `Wo` that implements both these interfaces.

**Part (d) [1 mark]**

Assume we introduced this new declaration in `EstasLoco#main()`:

`Wo nini = new Wo();` //where the class `Wo` is the one you defined in Part (c)

Why does the object `nini` not suffer from same ambiguity as the object `mimi`?

**Part (e) [4 marks]**

From the following list of declarations/instantiations, **circle** the ones that are **not** correct.

`Object a = new Wo();`      `IKo b = new Object();`      `IKo c = new IKo();`

`Object d = new ILo();`      `Wo e = new Mo();`      `ILo f = new Wo();`

**Part (f) [3 marks]**

Consider the following code:

```
public class Pyn {  
    public void doobi(){  
    }  
}
```

```
public abstract class Seb extends Pyn{  
    public abstract void yada();  
}
```

```
public class Zap extends Seb{  
  
}
```

```
public class NoLoco {  
    public static void main(String args[ ]){  
        Pyn pyni = new Zap();  
        pyni.yada();  
        Pyn pynk = new Seb();  
        System.out.println(pynk);  
        Pyn pyna = new Zap();  
        System.out.println(pyna.doobi());  
    }  
}
```

The above code has three errors (that are not syntax errors). List them in point form below.

**Question 5** [10 marks]

Consider this very simple stack class. Method bodies have been omitted.

```
public class Stack {  
    /**  
     * Construct an empty stack.  
     */  
    public Stack() {  
    }  
  
    /**  
     * Make x the new top item on this stack.  
     *  
     * @param x the new top item.  
     */  
    public void push(Object x) {  
    }  
  
    /**  
     * Remove and return the top item on this stack.  
     *  
     * @return the top item.  
     */  
    public Object pop() {  
    }  
}
```

Below is the JUnit code that is automatically generated for it:

```
public class StackTest {  
    public StackTest() {  
    }  
  
    @BeforeClass  
    public static void setUpClass() throws Exception {  
    }  
  
    @AfterClass  
    public static void tearDownClass() throws Exception {  
    }  
  
    @Before  
    public void setUp() {  
    }  
  
    @After  
    public void tearDown() {  
    }  
}
```

```
    }

    /**
     * Test of push method, of class Stack.
     */
    @Test
    public void testPush() {
        // Body omitted
    }
    /**
     * Test of pop method, of class Stack.
     */
    @Test
    public void testPop() {
        System.out.println("pop");
        Stack instance = new Stack();
        Object expectedResult = null;
        Object result = instance.pop();
        assertEquals(expectedResult, result);
    }
}
```

**Part (a) [3 marks]**

In the space provided below, rewrite the code for method `testPop()` so that it tests the behaviour of `pop()` on a stack with one element.

**Part (b) [3 marks]**

If we run these tests (testPush() and testPop()):

How many times will setUpClass() be called? \_\_\_\_\_,

and how many times will tearDown() be called? \_\_\_\_\_,

and how many times will setUp() be called? \_\_\_\_\_.

**Part (c) [4 marks]**

Suppose you wanted to create an object named "fixture" of type Stack to be available and in the following state for **every** test case: a stack of three integers, 10, 20 and 30, with 10 on the bottom and 30 on the top. Modify the code above to make this happen.

## Question 6 [10 marks]

### Part (a) [3 marks]:

Fill in the missing word:

- In the Java Collection Framework: We can use \_\_\_\_\_ when we need a collection that does not preserve order and does not allow repetition.
- In Java Collection Framework: We can use \_\_\_\_\_ when we need a collection that preserves order and allows repetition.
- In Java Collection Framework: We can use \_\_\_\_\_ when we need a collection that retrieves a value based on its key.

### Part (b) [2 Marks]:

Consider this List declaration: *List list = new ArrayList();* Suppose that this list only contains objects of type *String*. Write a "For loop" that prints the length of each String in the list.

You may need to use the following API's:

- *s.length()*; returns the length of a String *s*.
- *l.size()*; returns the size of List *l*.

### Part (c) [1 Mark]:

Rewrite the declaration of the *List list* in Part (b) to restrict it to store only String objects.



**Part (d) [2 marks]:**

Given the new declaration you came up with in Part c, rewrite the “For loop” you wrote in Part b. Your code should be different and you should not explicitly need the list size.

**Part (e) [2 marks]:**

Consider the following collection of type Set, “*Set set = new HashSet()*”. Assume it contains three objects of type String. What is wrong in the following “For loop” that is supposed to print the length of each string in the set?

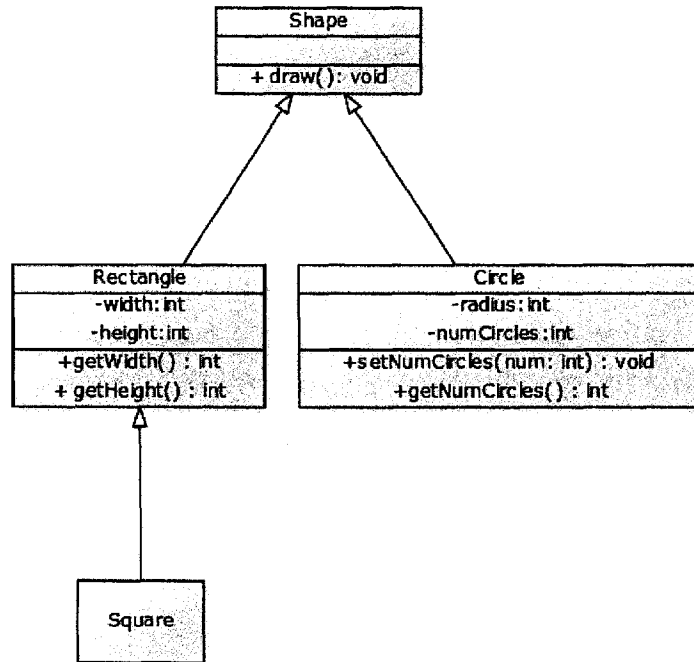
```
for (int i = 0; i < 3; i++){  
    String s = (String) set.get(i);  
    System.out.println(s.length());  
}
```

**Part (f) [Bonus 2 marks]:**

Fix the code in Part (e) above.

**Question 7** [10 marks]

The diagram below shows a UML class diagram for 4 classes. Answer the following questions based on the class diagram.

**Part (a)** [6 marks]

Write the equivalent Java classes. Write method return type and signatures and don't write method bodies.

**Part (b) [2 marks]**

One SOLID principle is violated in this design. Mention the name of the principle, and specify where it is being violated.

**Part (c) [2 marks]**

Assume that the two methods in Circle are the setter and getter of numCircles.

```
c1 = new Circle();  
c2 = new Circle();  
c3 = new Circle();  
c1.setNumCircles(1);  
c2.setNumCircles(2);  
c3.setNumCircles(3);
```

What will be the output of?

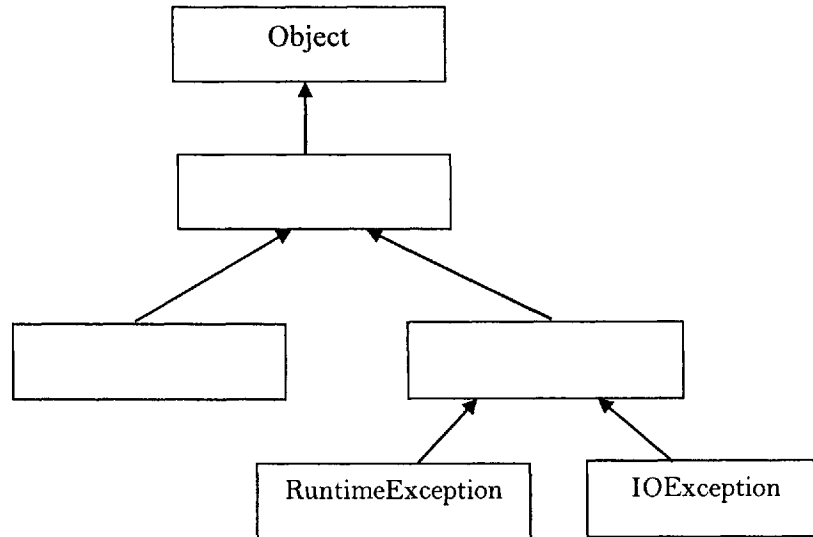
```
System.out.println(c1.getNumCircles());
```

Circle one of the following:

- (a) 0
- (b) 1
- (c) 2
- (d) 3

**Question 8** [10 marks]

Fill in the class hierarchy below with the names of the missing classes. Then fill in the blanks below.



RuntimeException, \_\_\_\_\_ and their subclasses are considered \_\_\_\_\_ exceptions.

All subclasses of \_\_\_\_\_ except for RuntimeException and its subclasses are considered \_\_\_\_\_ exceptions.

**Question 9** [10 marks]

Consider the following code for a class that is a **Singleton**, fill in each space with exactly one word.

```
public class Logger {  
    ① _____ ② _____ ③ _____ instance;  
  
    // Constructor  
    ④ _____ ⑤ _____ () {  
    }  
  
    ⑥ _____ ⑦ _____ ⑧ _____ getInstance() {  
        if(instance == null) {  
            instance = ⑨ _____ Logger();  
        }  
        return ⑩ _____;  
    }  
}
```

## Question 10 [12 marks]

### Part (a) [1 mark]

List the strings that can be matched by this regular expression:  $a[b-d]^+$

### Part (b) [1 mark]

Write **how many** strings can be matched by this regular expression:  $ab?[c-e]^+$

### Part (c) [2 marks]

Circle the strings that **cannot** be matched by this regular expression:  $(a?b[cd]^+)^+$

abcb      bbbbd      b      (the empty string)      bbabb      aabaa      abcabcabc

### Part (d) [2 marks]

Write a regular expression that accepts all the strings that can be created using only 0's (zeroes) and 1's (ones) that start and end with 0 (zero).

### Part (e) [2 marks]

Write a regular expression that accepts all the strings that can be constructed by writing at least two A's followed by at least three B's.

### Part (f) [4 marks]

Circle the regular expressions that **can** match the following line:

**9. The film format must be Academy 35 mm.**

$^9.(\backslash s \backslash w^*)+ \$$        $^ \backslash w. * \$$        $^ .. [ ^ x z q ] * \$$        $^ [ \backslash s \backslash w \backslash . ] * \$$        $^ . 35 . \$$

$^ \backslash d . ( ( \backslash w ) + ) * . \$$        $^ \backslash w + \$$        $^ 9 . \backslash s \backslash w \backslash s \backslash w \backslash s \backslash w \backslash s \backslash w \backslash s \backslash w \backslash s \backslash d \backslash d \backslash s m m . \$$

Use the space below for rough work. This page will not be marked, unless you clearly indicate the part of your work that you want us to mark.

Use the space below for rough work. This page will not be marked, unless you clearly indicate the part of your work that you want us to mark.



Use the space below for rough work. This page will not be marked, unless you clearly indicate the part of your work that you want us to mark.

Total Pages = (26)  
Total Marks = (100)