CPSC 101 Winter 2016
Midterm I—05 February 2016

Name (Printed) : _____

Signature : _____

Student Number : 2 3 0 _ _ _ _ _ _

Question	Score
1	/10
2	/6
3	/2
4	/2
5	/5
6	/5
7	/6
8	/5
9	/5
10	/8
11	/2
12	/2
13	/2
Total	/60

- This is a **50** minute exam. This exam contains **9** pages of questions not including this cover page. Make sure that you have all of them.
- Put your name on the top right hand corner of each page as examination papers sometimes come unstapled.
- *Read each question carefully. Ask yourself what the point of the question is. Check to make sure that you have answered the question asked.*
- Answer all questions on the exam sheet. If you do some of your work on the back of a page, clearly indicate to the marker what work corresponds with which question.
- Partial marks shall be awarded for clearly identified work.
- Non-programmable calculators and simple wrist-watches are allowed. **No cell-phones or other non-medical electronic devices**.
- This exam counts as **15%** of your total grade. There are **60** points total on the exam.

True and False

1 each

1. Circle **TRUE** or **FALSE** as appropriate. Questions that don't clearly indicate *one* choice shall be marked wrong. If you feel that the answer depends on how you interpret the question, give a brief reason for the answer you chose.

(a) A non-`static` method automatically has access to the `static` variables of the same class. **TRUE FALSE**

(b) In a static method, member variables can be accessed using the `this` keyword. **TRUE FALSE**

(c) Suppose that `fred` and `bill` are variables of the same type. The assignment "`fred = bill`" is illegal if the type of `fred` is an immutable class. **TRUE FALSE**

(d) The choice of which overriding method to call is made at compile time. **TRUE FALSE**

(e) An object of a class may access the private member variables of another object in the same class. **TRUE FALSE**

(f) A JAVA class may implement multiple interfaces. **TRUE FALSE**

(g) A JAVA class may extend multiple classes. **TRUE FALSE**

(h) It is illegal to subclass from `java.io.Writer`. **TRUE FALSE**

(i) It is illegal to subclass from `java.lang.String`. **TRUE FALSE**

(j) Considered as sets, superclasses contain less objects than subclasses. **TRUE FALSE**

Packages and Code Execution

(2) **2.** (a) What is the purpose of JAVA packages?

(2) (b) What file does JAVA load when the command

```
$ java mocha.version1.ScheduleHelper
```

is run at the command line?

(2) (c) Consider the `ScheduleHelper.java` file connected with the example above. What package statement does it contain? (Give the exact syntax.) Where is it located?

(2) **3.** Suppose that the command line

```
$ java mocha.version1.ScheduleHelper CSSchedule.txt
```

is executed. How can the `ScheduleHelper` program access the fact that `CSSchedule.txt` was specified on the command line?

(2) **4.** What are `.jar`-files? How are they created, and what is their purpose?

Object Oriented Design

(2) **5.** (a) What is a side effect?

(3) (b) Should side effects occur in methods with void return type, or non-void return type? More generally, what are some good practices in dealing with side-effects?

(2) **6.** (a) What is an immutable class?

(1) (b) Give an example of an immutable class in the standard JAVA library.

(2) (c) Explain how you would code a class so that it is immutable.

```
1  public class BankAccount
2  {
3    private double myBalance ;
4
5    public BankAccount()
6        {
7        myBalance = 0.0 ;
8        }
9    public void deposit(double amount)
10       {
11       // ...
12       }
13   public void withdraw(double amount)
14       {
15       // ...
16       }
17   public double getBalance()
18       {
19       return myBalance ;
20       }
21 }
```

Figure 1: Sample BankAccount code for Question 7

(2) **7.** (a) What are *pre-conditions* and *post-conditions*?

(2) (b) What is a likely pre-condition of the `withdraw` method shown in Figure 1?

(2) (c) What is a likely post-condition of the `withdraw` method shown in Figure 1?

Inheritance

(5) **8.** Write a `SpecialSavings` account that is a subclass of `BankAccount` shown in Figure 1 on the preceding page.

The `SpecialSavings` account should have one new method

```
void payInterest()
```

that adds 1% of the current balance to the account.

It should also override the `withdraw` method so that the user is charged an additional $0.15 if the account balance is less than $500.00.

9. Suppose that `CanadianDollars` is a direct subclass of `Dollars`, which in turn is a sub-class of `Currency`.

(1) (a) How would you draw this using a UML diagram?

Suppose further that we are looking at the code:

```
1  Currency debt = new CanadianDollars(3400.15) ;
2  // ...
3  System.out.println(debt.value()) ;
```

(1) (b) In which class must the `.value()` method exist in order for the code to compile?

(1) (c) Suppose that no assignments to `myDebt` happen between lines 1 and 3, and suppose that the `Currency`, `Dollars`, and `CanadianDollars`

classes all have `.value()` methods. In this case, which class's `.value()` method is executed at run-time?

(2) (d) Again, suppose that no assignments to `myDebt` happen between lines 1 and 3, and suppose that the `Currency` class and the `Dollars` class have explicit `.value()` methods, but that the `CanadianDollars` class does not.

What happens at line 3 at compile- and/or run-time in this circumstance? Justify your answer.

Coding Question

(8) **10.** Suppose that objects of the `DriversLicense` class each have a unique integer serial number. To implement this, the `DriversLicense` class has two private long integer member variables `theNextDriversLicenseNumber` and `myDriversLicenseNumber`.

(a) Which of these are likely to be static, and which are likely to be non-static? Why

(b) Write a zero-argument constructor for `DriversLicense`s that causes each object to have a unique serial number. You may assume that there are far less `DriversLicense`s than there are positve `long` integers.

(c) Also write an accessor method `getNumber` that returns this `DriversLicense`'s serial number.

(d) What effect would making the `myDriversLicenseNumber` variable `final` have? Would you advise this?

(e) What effect would making the `getNumber` method `final` have? Would you advise this?

Interfaces

(2) **11.** What is an abstract method? How does an abstract method differ syntactically from a concrete (non-abstract) method?

(2) **12.** Suppose that the interface `SlackWorthy` promises a single method

```
void doNothing(Object art) ;
```

Write code for a (very small) class that implements `SlackWorthy`.

(2) **13.** Write the code for an interface `Runnable` that promises a single method `public void run()`.