

CPSC 101 Winter 2017
Midterm I—03 February 2017

Name (Printed) : _____

Signature : _____

Student Number :

2	3	0					
---	---	---	--	--	--	--	--

Question	Score
1	/10
2	/6
3	/6
4	/2
5	/1
6	/2
7	/2
8	/5
9	/3
10	/5
11	/7
12	/7
13	/2
14	/2
Total	/60

- This is a 50 minute exam. This exam contains 8 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`" destroys the object that `fred` was pointing at. **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.lang.String`. **TRUE FALSE**
 - (i) Considered as objects, superclass objects contain less than subclass objects. **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.ScoreIV
is run at the command line?
- (2) (c) Consider the ScoreIV.java file connected with the example above.
What package statement does it contain? (Give the exact syntax.)
Where is it located?

Object Oriented Design

- (6) 3. This question is about side effects and naming.
- (a) What is a side effect?

- (b) JAVA collections like, for example, `ArrayList<String>` have a `getIterator()` method that returns an `Iterator`. JAVA Iterators have a non-void `next()` method that returns the next object in the collection and advances the iterator. In terms of naming and side effects, is this good software engineering? Why or why not?
- (c) Other programming languages have `Iterator`'s with a slightly different interface. There is a `current()` method that returns the value that the iterator is currently pointing at, and a void `advance()` method that moves the iterator to the next object. In terms of naming and side effects, is this good software engineering? Why or why not?
- (2) 4. What are *cohesion* and *coupling* and how do they relate to good class design?
- (1) 5. What is the difference between an attribute and state?
- (2) 6. What are behaviours and collaborations?

JAVA Memory Organization

7. Speaking of JAVA,

(1) (a) what causes an object to be created in memory?

(1) (b) What causes an objects to be removed from memory?

(1) 8. (a) When are stack frames created?

(1) (b) When are stack frames destroyed?

(2) (c) Name two kinds of information stored in a stack frame.

(1) (d) What is the difference between the stack frame corresponding to a static method and the stack frame corresponding to a non-static method?

9. This question is about the regions of a JAVA program's memory.
- (1) (a) What is the name of the region of memory that stores variables and arguments to functions?
- (1) (b) What is the name of the region of memory that stores objects?
- (1) (c) What is a third region of memory used by a running JAVA program.

Inheritance

- (1) 10. (a) Explain very briefly why an assignment like
- ```
1 Object obj = "catFish" ;
```
- is legal.
- (2) (b) Write a very short code fragment that uses `instanceof` to determine whether or not an `Object` variable `obj` is really a `String`.
- (2) (c) Even in an `Object` variable `obj` is really a `String`, the method call `obj.length()` is illegal. Show how to use a down-cast to get the length of the string "in" `obj` in this situation.

---

```
1 public class BankAccount
2 {
3 private double myBalance ;
4
5 public BankAccount() { myBalance = 0.0 ; }
6 public void deposit(double amount) { /* ... */ }
7 public void withdraw(double amount) { /* ... */ }
8 public double getBalance() { return myBalance ; }
9 }
```

---

Figure 1: Sample BankAccount code for Question 11

- (7) **11.** Write a `SpecialSavings` account that is a subclass of `BankAccount` shown in Figure 1.

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.

- (7) 12. A `DriversLicense` class has the following specification.
- (i) Each driver's licence has a unique (long) serial number, accessible through a method named `getLicenseNumber()`.
  - (ii) code like

---

```
DriversLicense d = new DriversLicense() ;
System.out.println(d)
```

---

prints something like "BC Licence: 31241". (For examination purposes, the format of the numeric part does not matter.)
  - (iii) The class has an explicit zero-argument constructor.

---
- (a) Write the code for this class.

(b) What effect would making the non-static member variables `final` have? Would you advise this?

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

## Interfaces

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

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

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

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