

CPSC 101 Winter 2013  
Midterm I—01 February 2013

Name (Printed) : \_\_\_\_\_

Signature : \_\_\_\_\_

Student Number : 

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

---

Question	Score
1	/10
2	/2
3	/2
4	/2
5	/2
6	/8
7	/3
8	/4
9	/5
10	/3
11	/3
12	/6
13	/5
14	/6
15	/4
Total	/65

- 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 **65** 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) When a method returns, all local variables are automatically destroyed and their storage is recovered. **TRUE FALSE**
  - (b) A non-static method automatically has access to the static variables of the same class. **TRUE FALSE**
  - (c) When a method returns, all objects it created are destroyed and their storage is recovered. **TRUE FALSE**
  - (d) When a non-static method is called the compiler generates a hidden argument which is a reference to the object calling the method. **TRUE FALSE**
  - (e) Suppose that `fred` and `bill` are variables of the same, immutable type. Then the assignment "`fred = bill`" is illegal. **TRUE FALSE**
  - (f) The assignment "`fred = bill`" causes both `fred` and `bill` to point at the same object. **TRUE FALSE**
  - (g) For a given class name, method signature combination there is exactly one chunk of code memory. **TRUE FALSE**
  - (h) An object of a class may not access the private member variables of another object in the same class. **TRUE FALSE**
  - (i) A `JAVA` class may implement multiple interfaces. **TRUE FALSE**
  - (j) A `JAVA` class may extend multiple classes. **TRUE FALSE**

## Interfaces

- (2) 2. What is an abstract method? How does an abstract method differ syntactically from a concrete (non-abstract) method?
- (2) 3. What kind of variables can an interface contain?
- (2) 4. The interface `Comparable<Time>` promises a single method
- ```
int compareTo(Time t) ;
```
- What must a `Time` class do in order to implement the `Comparable<Time>` interface?
- (2) 5. The `Runnable` interface promises a single void method `run()`. Write the code for the `Runnable` interface.

## Packages

- (2) 6. (a) What file does JAVA load when the command  
    `java crossword.Grid`  
    is run?
- (2) (b) How does the CLASSPATH variable affect this?
- (2) (c) In the `Grid.java` file connected with the command above, what  
    package statement is there? Where is it located?
- (2) (d) What class(es) must be in the `Grid.java` file connected with the  
    command above? What method(s) must be in the `Grid.java` file?
- (3) 7. When we use a class name in code when can we use the short name

(for instances `Scanner`) and when do we need to use the long name (for instance `java.util.Scanner`)? Be as precise as you can about the rules.

- (2) 8. (a) What are `.jar`-files? How are they created, and what is their purpose?
- (2) (b) How do you use `jar`-files once you have created them?

## Memory Organization

- (1) 9. (a) When are stack frames created?
- (1) (b) When are stack frames destroyed?
- (2) (c) Name two different 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?
- (3) 10. What information is stored on the heap? How is it created? When is it destroyed?

## Object Oriented Design

- (3) 11. Contrast cohesion and coupling. Which is desirable? Which involves multiple classes?
- (2) 12. (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?
- (1) (c) How do side effects relate to Attributes, Behaviours, and Collaborations?
- (2) 13. (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.

- (2) 14. (a) What are *pre-conditions* and *post-conditions*?
- (2) (b) What is a likely pre-condition of the `deposit` method shown in Figure 1?
- (2) (c) What is a likely post-condition of the `deposit` method shown in Figure 1?

---

```
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 14

## Coding Question

- (4) 15. Suppose that objects of the `Widget` class each have a unique integer serial number. To implement this, the `Widget` class has two private integer member variables `theNextSerialNumber` and `mySerialNumber`.
- (a) Which of these are likely to be static, and which are likely to be non-static?
- (b) Write a zero-argument constructor for `Widgets` that causes each object to have a unique serial number. (Writing other code for the class is *not* necessary.)