CPSC 101 Winter 2012
Midterm I—03 February 2012

Name (Printed) : _____

Signature : _____

Student Number : 2 3 0 | | | | | | |

Question	Score
1	/13
2	/4
3	/4
4	/2
5	/5
6	/3
7	/5
8	/4
9	/10
Total	/50

- This is a **50** minute exam. This exam contains **7** 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 **50** points total on the exam.

Memory Organization

(4) **1.** (a) Name the parts of the memory diagram shown in Figure 1 on the next page.[1]

- top-left _____
- top-right _____
- bottom-left _____
- bottom-right _____

(3) (b) Write a short chunk of code for the `main` method that might result in the situation shown.

(2) (c) Write the likely two-line code for the `Person.murder` method. Look at the diagram carefully to determine the order of execution of the statements.

(2) (d) Which of the methods that have been invoked but have not returned are static, and which are non-static? How can you tell?

[1] The figure pays tribute to two very famous computer scientists: Rear Admiral Grace Hopper (1906–1992) (COBOL) and Dennis Ritchie (1941–2011) (C). The figure is entirely fictional; both died of natural causes.

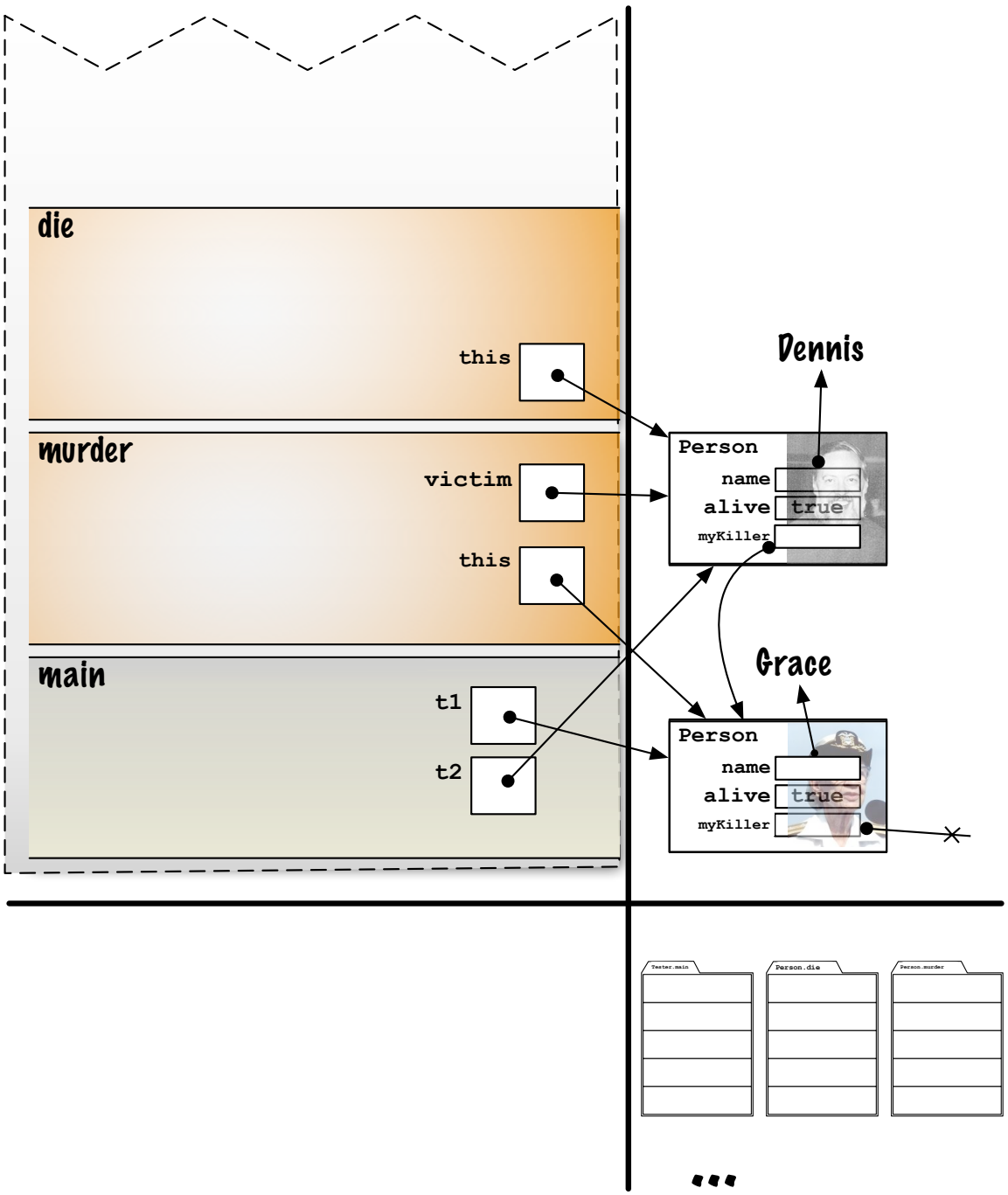

Figure 1: Memory diagram for Question 1 on the preceding page

(2) (e) Name a couple kinds of information that are in the stack frames but not shown in this figure.

Interfaces

2. Interfaces may contain both methods (member functions) and fields (member variables), but subject to very strict conditions.

(2) (a) The restriction on fields (member variables) is that they must be:

(2) (b) The restriction on methods (member functions) is that they must be:

Packages

(2) **3.** (a) If

```
/home/casper/Code/Lab19/crossword/Grid.java
```

is compiled to

```
/home/casper/Code/Lab19/crossword/Grid.class
```

and then run using

```
java crossword.Grid,
```

in what directory should the `java` command be issued?

(2) (b) Assuming that the `Grid.class` file is to be run as above, that is, as `crossword.Grid`, what should the `package`-statement in `Grid.java` be? Where must it be located?

(2) **4.** Explain as clearly as you can what the statement

```
import java.util.Scanner ;
```

does. How can you avoid using this import statement?

Longer answer

(3) **5.** (a) In practising Object Oriented Design, ABC's refer to Attributes, Behaviours, and Collaborations. Explain what these are.

(2) (b) Compare and contrast *private member variables* and *attributes*.

(3) **6.** What are cohesion and coupling? Explain the role that they play in class design.

(3) **7.** (a) Explain what a class invariant is.

(1) (b) When might a class invariant *not* hold?

(1) (c) In proving by induction that a class has a particular class invariant, constructors provide the _____.

Coding Questions

(4) **8.** Create a `SerialNumber` class that has one `public` zero-argument constructor, and two public methods

```
public long value() { ... }
public String toString() { ... }
```

that **return** an integer like `124357689` or a string like `"S/N: 124357689"` respectively. Every `SerialNumber` must have a unique `value()`. You may assume that less than one billion `SerialNumber` objects will be manufactured.

True and False

1 each

9. 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 name 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**