

OLLSCOIL NA hÉIREANN
THE NATIONAL UNIVERSITY OF IRELAND, CORK
COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

Summer Exam 2012

Second Year Computer Science

CS2500: Software Development

Professor Ian Gent
Professor J.A. Bowen
Dr M. R. C. van Dongen

INSTRUCTIONS: Answer **all 9** questions for full marks (225). All questions carry equal marks (25). There is **no need** to write whole classes *except where this is explicitly stated*. There is **no need** to write import statements. Use meaningful identifier names. Pay attention to the layout of your code and make sure your coding style adheres to the Java coding conventions. There is **no need** to add comments.

3 hours

Question 1: Basics.*(25/225 marks)***Question 1.a.***(10 marks)*

Provide a `main()` method that uses the enhanced for notation to print its arguments to standard output.

Question 1.b.*(15 marks)*

Provide a method that creates a `Scanner` to read from standard input. The method should use the `Scanner` to read all `ints` on standard input and print their sum to standard output. The method should be robust, so it should read integers until the user provides input that isn't an `int` or until there is no more input.

Question 2: Classes.*(25/225 marks)***Question 2.a.***(10 marks)*

What is a class and how does it relate to an object?

Question 2.b.*(15 marks)*

Implement a class called `Switch`. The purpose of the class is to represent an on-off switch.

A switch can be on or off and the class should provide a method called `getState()` that returns `true` if the switch is on and `false` if the switch is off. The class should also provide a method `toggleState()` that turns the switch off if it was on and vice versa. Initially, a switch should be on. Finally, the class should provide a method called `getSwitchCount()` that returns the number of `Switch` instances created so far.

Question 3: Inheritance.*(25/225 marks)***Question 3.a.***(10 marks)*

Explain the notion of inheritance, and state three advantages of inheritance.

Question 3.b.*(10 marks)*

Provide a concrete example of how you may implement inheritance in Java. Explain your example in terms of the notions of 'being more specific' and 'being more general', and relate these notions to the notions of 'is-a', 'extension', 'subclass', and 'superclass'.

Question 3.c.*(5 marks)*

Provide an example of overloading and an example of overriding. Explain the difference between overloading and overriding.

Question 4: Class Design.

(25/225 marks)

Using proper object-oriented class design, implement the classes in a class hierarchy for representing a Book in a Book shop. The following are the requirements.

- There are two kinds of Book categories: PaperBook and EBook.
- There are two PaperBook categories: Hardcover and Paperback.
- Each Book has a title, a page count, and a price.
- The price of an EBook is given by $\text{€ } 0.10 \times p$, where p is the number of pages of the EBook.
- The price of a PaperBook is given by $\text{€ } 0.15 \times p + c$, where p is the number of pages of the book and c the price of the book's cover.
- The cover price of a Paperback is given by $\text{€ } 1.00$.
- The cover of a Hardcover book costs $\text{€ } 5.00$.

To reduce the amount of work, there is no need to implement the class for Hardcover books. However, you may not exploit this to simplify your class design.

Question 5: Listeners.

(25/225 marks)

Provide a class with a `main()` method displaying a window with a button and a label in it. The initial size of the window should be 200×300 pixels. The button should display the text 'click me.' The text on the label should be the number of times the user has clicked the button. *Hint: you only have to implement one class that corresponds to a label that listens to the button.*

Table 1 on Page 6 lists some methods and constants which you may use to answer this question. Notice that the table is not exhaustive and lists some red herrings.

Question 6: Recursion.

(25/225 marks)

Question 6.a.

(10 marks)

Explain the notion of recursion.

Question 6.b.

(15 marks)

Implement a recursive method for squaring a non-negative int. The method is not allowed to use multiplication. *Hint: use the fact that $(x + 1)^2 = x^2 + 2 \times x + 1$ and find a way to avoid the multiplication.* **No marks are awarded if you provide an iterative solution or a solution that uses multiplication.**

You should provide a short explanation how your solution works: this should include a termination proof. This may be done in the form of a separate explanation or Java comments.

```
public class PartialIterableClass /* FILL IN #1 */ {  
    private String[] things;  
  
    public PartialIterableClass( String[] things ) {  
        this.things = things;  
    }  
  
    /* FILL IN #2 */  
}
```

Figure 1: Contrived partial class.

Question 7: Iterators and Generics.

(25/225 marks)

Question 7.a.

(10 marks)

Figure 1 depicts a contrived partial class which implements the `Iterable` interface. The only purpose of instances of this contrived class is to provide a mechanism to iterate over the members of the attribute `things`.

This class can be completed by providing code for the `/* FILL IN #1 */` and the `/* FILL IN #2 */` in Figure 1. To answer this question must complete this class. Make sure you clearly indicate which code completes `/* FILL IN #1 */` and which code completes `/* FILL IN #2 */`.

Question 7.b.

(15 marks)

Provide a *generic* class `Stack` for representing a stack with the following last-in-first-out API:

- o The method `push()` adds a new thing on top of the stack;
- o The method `pop()` returns the current thing on top of the stack.
- o The method `isEmpty()` returns `true` if and only the stack is empty.

Question 8: Exceptions.

(25/225 marks)

Joe uses his computer and his of `JoeThermometer` class to measure the temperature outside. The class provides the following constructor and instance methods.

```
JoeThermometer( ): Open a connection with the thermometer outside.  
double getTemperature( ): Get the current temperature.  
void close( ): Close the connection with the thermometer.
```

Last March Joe's class started throwing errors because of the violent solar storms.

Using **proper exception handling**, demonstrate how to use Joe's `JoeThermometer` class to (1) create a `JoeThermometer` instance, (2) use the instance to print the current temperature, and (3) close the connection with the thermometer. The exception handling should provide as many details as possible about the cause of failure, and print the stack trace which led to the failure.

Question 9: Enumerated Types.

(25/225 marks)

Question 9.a.

(10 marks)

Provide a critical comparison of `int` enums and `Java` enums. State advantages and disadvantages.

Question 9.b.

(15 marks)

Using `Java` enums, implement a class for the planets in our solar system. There is no need to create instances for all the planets. It suffices if you create instances for `Earth`, `Venus`, and `Mars`.

The class should provide getter methods for the mass and diameter of the heavenly bodies. You may use the following data for the mass and diameter.

```
Earth Mass:  $5.975 \times 10^{24}$  kg; diameter:  $6.378 \times 10^6$  m.  
Venus Mass:  $4.869 \times 10^{24}$  kg; diameter:  $6.052 \times 10^6$  m.  
Mars Mass:  $6.419 \times 10^{23}$  kg; diameter:  $3.393 \times 10^6$  m.
```