UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG
School of Electrical and Information Engineering
SD Methodologies, Analysis and Design

Object-Oriented Concepts — Presentation Topics

The time-limit for each presentation is strictly 20 minutes. Half of each group's members must present on OO Concepts and the other half on Design Patterns (still to be published). For a group of five members, three members will present on one topic and two on the other. All presenters for a particular topic will receive the same mark.

Presentations need to be *interactive*, in other words, the class needs to be involved in some *activity or exercise* to illustrate the topic that you are explaining. This implies that your actual talk-time is much less than 20 minutes!

Please email your presentations, in PDF or Powerpoint format, to me by 10:00 on the presentation day (joshua@jcse.org.za).

The following topics are intended to both highlight and revise fundamental object-oriented concepts and associated concepts. You may assume that your audience has a basic understanding of Object-Oriented Programming (OOP) which needs refreshing. Most topics have a number of questions which should be dealt with in your presentation, but your presentation should not be confined by these questions.

1 Memory Management

- Whose responsibility is memory management in C++, Java and C# (the programmer's or the run-time environment)?
- What is the difference between deterministic and non-deterministic object destruction?
- How is memory management handled in C++ *and* C# or Java?
- What is garbage collection?
- Give a code example in C++ **and** in Java or C#.

2 Value and Reference Semantics

- How are objects copied by default in different languages (C++, Java, C#) and is it possible to change this default behaviour?
- What is the difference between value types and reference types?
- What is the difference between passing by reference and passing by value and what are the implications?
- What does the 'new' keyword do (when instantiating a new object)?
- Give code examples to illustrate some of the above points.

3 OOP Basics

- What is a type?
- What is a class?
- What is an object?
- What does the *this* keyword refer to when used inside a class?
- What does the *static* keyword mean?
- What is inheritance?
- Give code examples of the above in C++, Java or C#.

4 Inheritance and Class Hierarchies

- Why is inheritance important?
- Explain single and multiple inheritance.
- Explain polymorphism and virtual functions.
- What are your thoughts on multiple inheritance? Which OO languages support it?
- Give a code example of single inheritance in C++, Java or C#.

5 Abstract Classes and Interfaces

- What is the difference between an interface and an abstract class?
- Why are abstract base classes and interfaces so important and how are they used?
- How do you decide whether to use an abstract base class or an interface?
- What is polymorphism?
- Explain the Liskov Substitution Principle
- Give a code example in C++ **and** in Java or C#.

6 Features of Modern Object-Oriented Languages 1

Explain *two* of the following language features; illustrate with code examples and typical usage scenarios:

- Reflection (C# or Java)
- Delegates (C#)
- Method references (Java)
- Extension methods (C#)
- Default Interace method implementations (Java)

7 Features of Modern Object-Oriented Languages 2

Explain *two* of the following language features; illustrate with code examples and typical usage scenarios:

- Lambdas (C# or Java)
- Anonymous types (C#)
- Anonymous classes (Java)
- Dynamic object/Type (C#)

8 Generic Programming

- What is generic programming and how is it different to using inheritance and interfaces?
- In which situations is it appropriate to use generics?
- Does generic code imply or enforce an interface on the instantiating types?
- Give a code example in C++ **and** in Java or C#.