CET325

Advanced Mobile Development

Session 1A

Learning Objectives

- Module Overview
  - Understanding module structure
  - Identify indicative content
  - Introduce assignment schedule

- Java Primer:
  - Introduce Java Programming Language
  - Evaluate differences between Java and C#

Module Overview

- Delivery method
  - Lecture (2 hours)
    - We describe and present content
    - We propose problems and challenges
    - You can ask questions, take notes, and learn/think
  - Tutorial (2 hours, plus independent study)
    - You practice/experiment and ask for help
    - We suggest solutions, give feedback and support

Module Overview

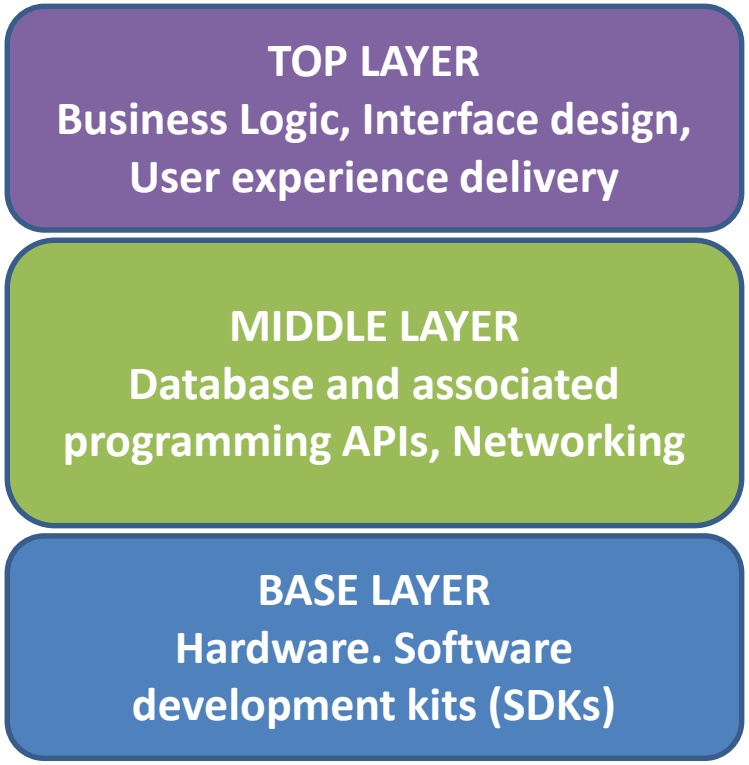
- Module Team:
  - Paolo Modesti (leader) paolo.modesti@sunderland.ac.uk
  - Andrew Smith andrew.smith@sunderland.ac.uk

Module Overview

- **Module Aims:**
  - Enable students to build native mobile applications.
  - Understand the systems involved (hardware, software, database technologies, interface features)
  - Adopt an integrated and tiered approach which conforms to design and data structure standards of the designated mobile platform.

Module Overview

- Content will be structured via a 3-tiered paradigm.



Module Overview

- Assessment - You will have a single assessment which contributes 90% of the final module mark.
  - Final Submission – Friday 13th January 2017, midnight
- Tutorials: The practical exercises are designed to help you develop the necessary skills to successfully complete the module.
  - It is imperative that you stay on track with these every week.
- Engagement: 10% of the final mark (tutorial submissions)

Engagement

- Tutorial: upload periodically your work on SunSpace.
- Typical deadline: for week *n*, it will be before the first lecture in week *n+1* (Tuesday midday).
- 1st deadline: Tue 4th October
- Submission minimal requirements:
  - Original/Genuine
  - Pertinent
  - No formal assessment/marking of each exercise
- Rationale: self-assessment, demonstrate your engagement and provide feedback to the instructors

Programming Languages

Do you know any of the following programming languages?

- C
- C++
- C#
- Python
- Java Script
- Java
- Haskell
- OCaml
- Scala
- …

Introduction to Java

- The Java programming language is a high-level programming language

Simple	Architecture neutral
Object oriented	Portable
Distributed	High performance
Multithreaded	Robust
Dynamic	Secure

The Java Language Environment
white paper by James Gosling and Henry McGilton (1996)

Object Oriented Programming Revision

Concept	Description
Class	1. An instance of a class with its own identity
Object	2. A variable with “deep copy” behaviour
Method	3. A property or method embedded within a class
Member	4. A partial blueprint of a class, too incomplete to be instantiated into an object
Value Type	5. A function defined in a class
Reference Type	6. The scope of accessibility for a class’s member
Public / Private / Protected	7. The blueprint of an object, defining its properties and behaviours
Interface	8. A variable with “shallow copy” behaviour

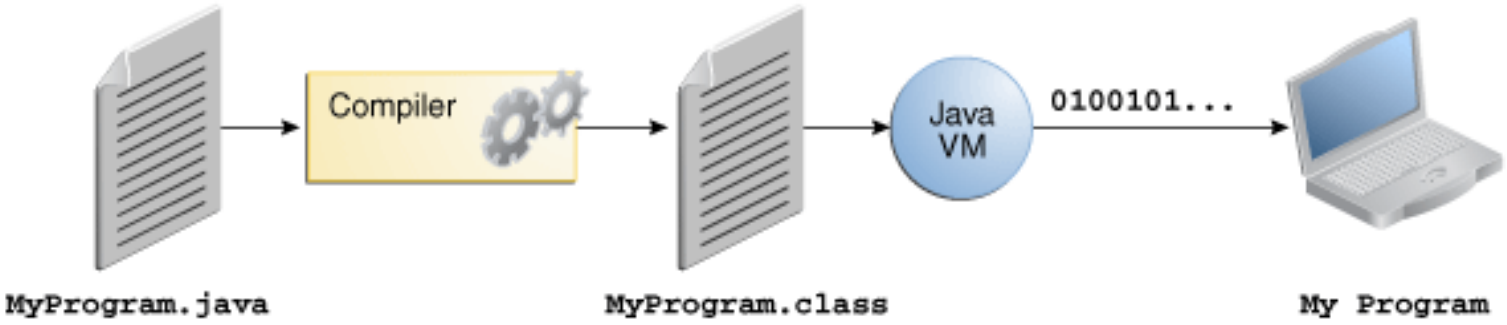
- *Match the common OO concepts / keywords to their descriptions*

Object Oriented Programming Revision

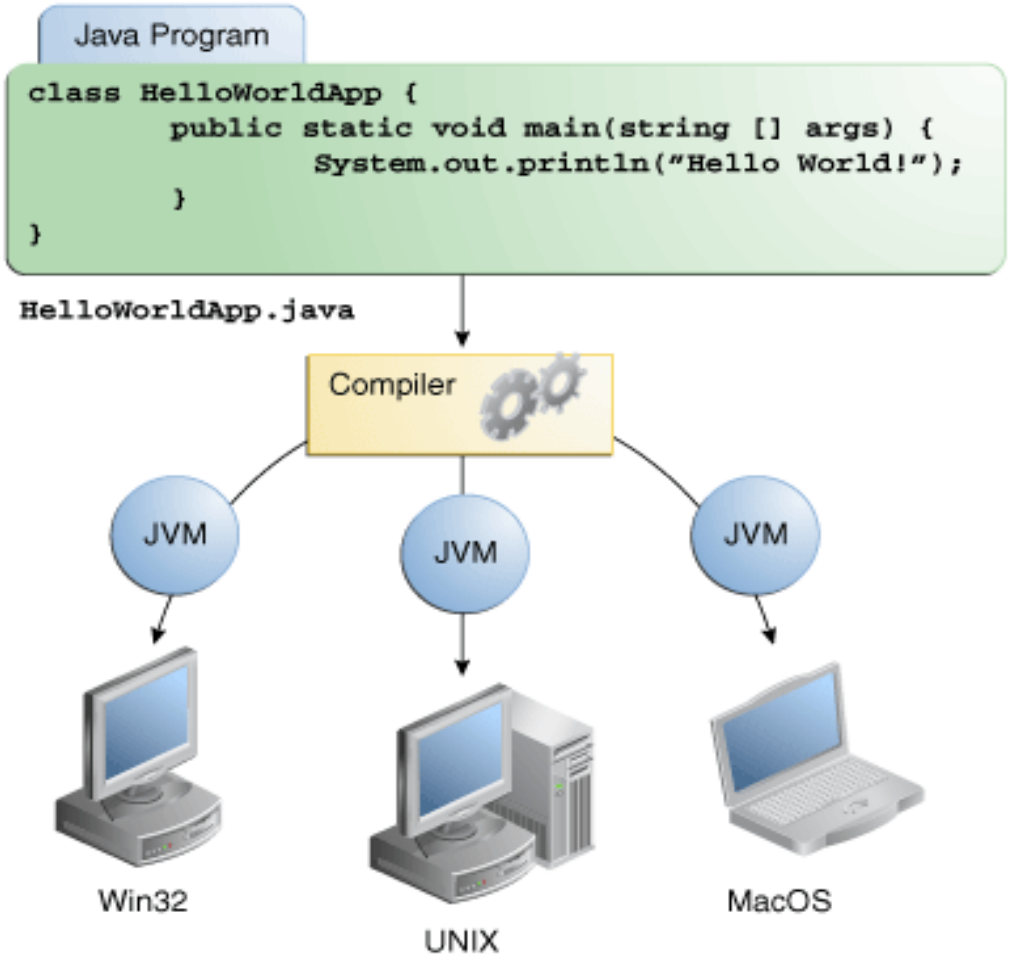
Concept	Description
Class	7. The blueprint of an object, defining its properties and behaviours
Object	1. An instance of a class with its own identity
Method	5. A function defined in a class
Member	3. A property or method embedded within a class
Value Type	2. A variable with “deep copy” behaviour
Reference Type	8. A variable with “shallow copy” behaviour
Public / Private / Protected	6. The scope of accessibility for a class’s member
Interface	4. A partial blueprint of a class, too incomplete to be instantiated into an object

- *Match the common OO concepts / keywords to their descriptions*

Compiling and Running Java Programs

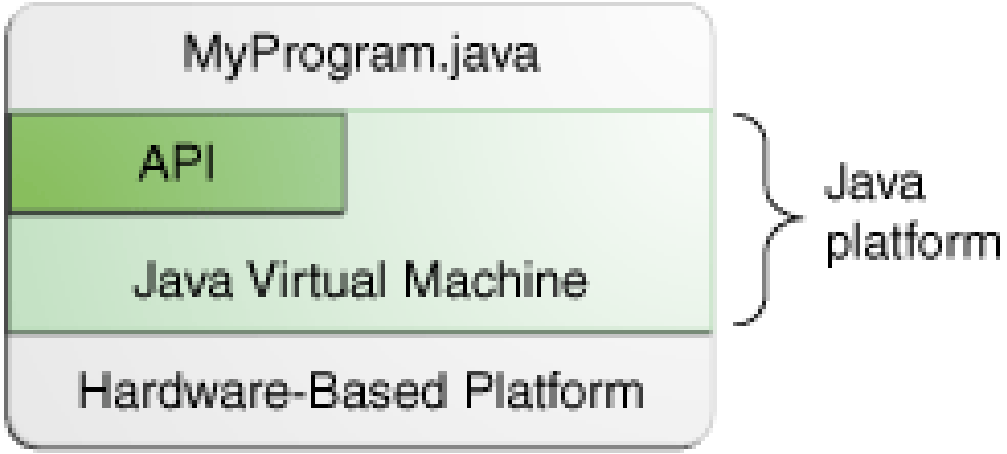


Compiling and Running Java Programs



Java Platform

- The Java platform is a software-only platform that runs on top of other hardware-based platforms.



Java – In summary

- Java is a platform-independent language
  - The output of Java compiler is NOT executable!
  - It's bytecode.
  - This bytecode is then executed by a Java Virtual Machine (JVM), which interprets the byte code.

Hello World!

```
public class HelloWorld {
        public static void main(String[] args) {
            System.out.println("Hello World");
        }
}
```

- Class declaration
- Main method declaration
- Variable

Tutorial

- Tutorials – Terraces A-B