Introduction to Module

Content, Structure and the JVM

Produced by: Dr. Siobhán Drohan
Ms. Mairead Meagher

SE TU
Ollscoil
Teicneolaíochta
an Oirdheiscirt

South East
Technological
University

Department of Computing and Mathematics
http://www.setu.ie/

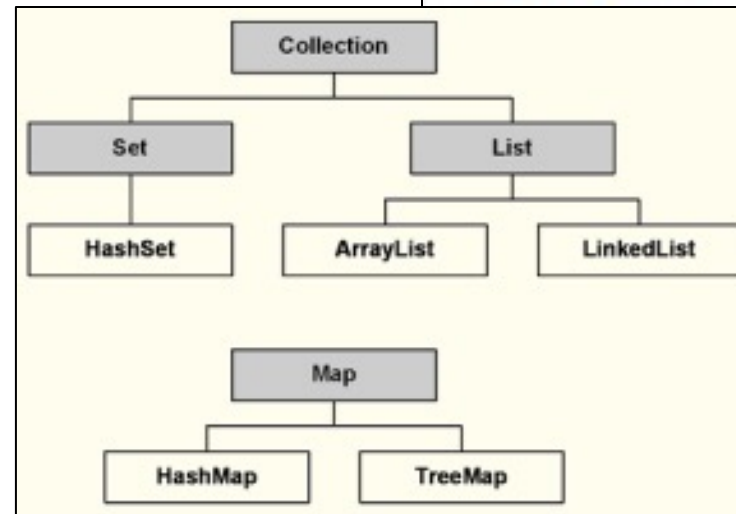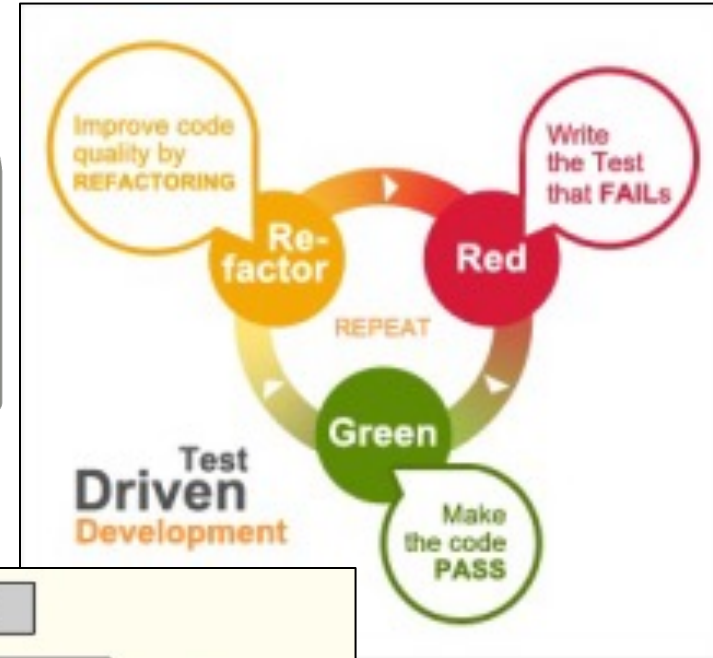# Topics list

1. Module Content and Structure.

2. Java Language:

   a) **Files** in Java.

   b) Java Virtual Machine (**JVM**).

   c) **main()** method.

Module Content

Module Content

- 100% CA:
  - Assignment 1
  - Assignment 2

- Focus on:

  - Menu Driven Systems (console based)

  - Data Structures

  - CRUD and Persistence (XML)

  - Exceptions

  - Inheritance, Polymorphism and Abstraction

Topics list

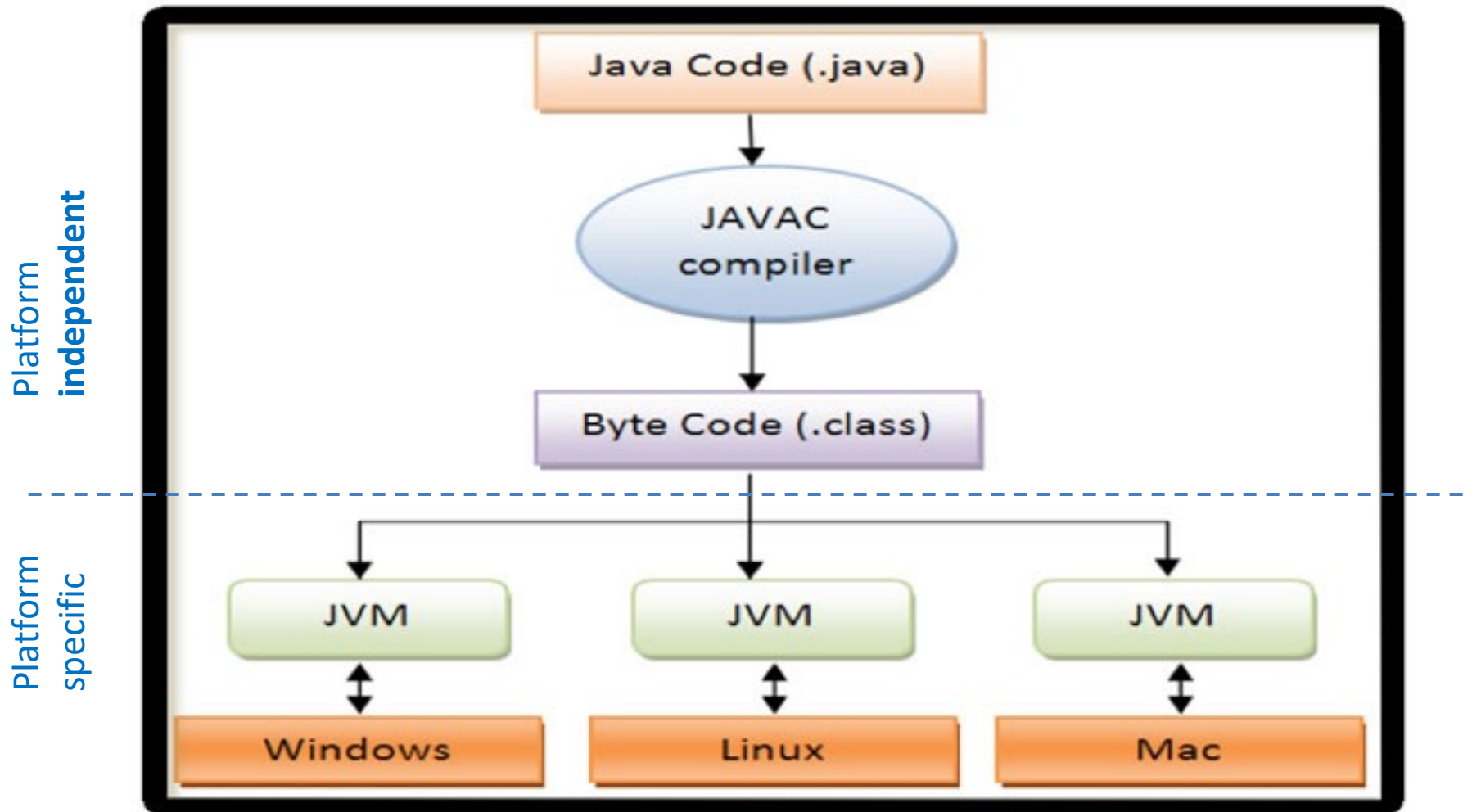1. Module Content and Structure.

2. Java Language:

   a) **Files** in Java.

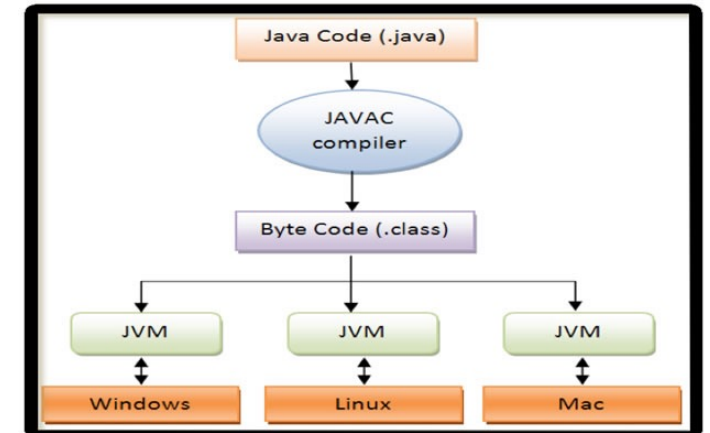   b) Java Virtual Machine (**JVM**).

   c) **main()** method.

Files in Java (1)

Files in Java (3)

- **Java code** is written in **.java** file.
  - code contains one or more Java language constructs e.g. Classes, Methods, Variable, Objects etc.
- **Compiling** this code generates a **.class** file
    - A **.class** file
      - "**byte code**"
      - the input to Java Virtual Machine (JVM)
      - The **JVM** *(for a specific platform)*
        - **reads** this byte code
        - **interprets** it as machine code instructions *(for that platform)*
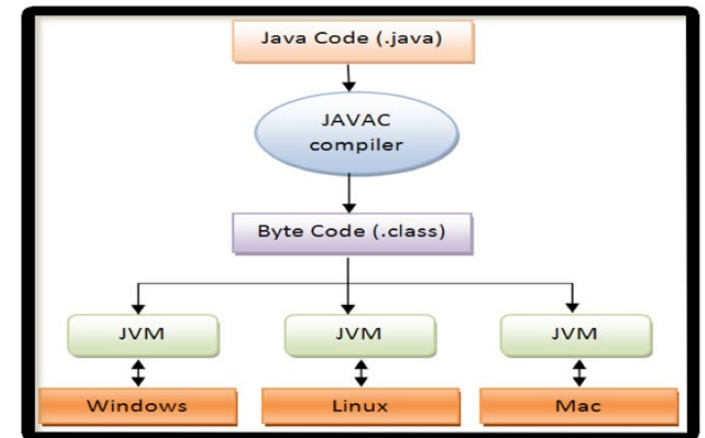   - and **executes** the program *(on that platform)*

Topics list

1. Module Content and Structure.

2. Java Language:
   a) **Files** in Java.
   b) Java Virtual Machine (**JVM**).
   c) **main()** method.

Java Virtual Machine (JVM)

- A "**virtual**" computer that resides in the "real" computer as a software process.

- Gives Java the flexibility of **platform independence**.

- The **.class** files can be run on any OS, once a JVM has been installed
  - (NB: JVM is installed when you install the JDK).

Topics list

1. Module Content and Structure.

2. Java Language:
   a) **Files** in Java.
   b) Java Virtual Machine (**JVM**).
   c) **main()** method.

main() method

- When you want to run a java project, the Java Virtual Machine (JVM) invokes the **main()** method in the project.

- For the JVM to recognise it, the main() method <u>must</u> have a specific method **signature**.

```
public static void main(String[] args)
{
    ...
}
```

main() method - signature

- "main" must exist
- "main" must be **public**
- "main" must be **static** (class method)
- "main" must have a **String array parameter**
- Only "main" can be invoked automatically.

```
public static void main(String[] args)
{
    ...
}
```

The **main()** method typically...

```
public static void main(String[] args)
{

    Driver driver= new Driver();
    driver.runMenu();

}
```

Creates an object

Calls the first method

Questions?