

OOP

CSA 1012 Object-Oriented Programming

Mr. Joseph Cordina
Rm 203, New Comp. Bldg.
E-mail:
joseph.cordina@um.edu.mt



OOP

Course Objectives

- Familiarity with the syntax of the Java or C# language
- Programming in an object oriented style
- Building structured, reusable, manageable platform independent stable code
- Course Structure:
 - Basic Java/C# syntax
 - OOP
 - OOP using Java/C#



Introduction

- 2 Hour lecture will be given every week
- You are assumed to **know** previous lecture by the next lecture.
- Practicing at home will be necessary since organised practical sessions will be difficult to organise.
- Course Notes will be supplied and are available at:
 - www.cs.um.edu.mt/~jcord
- Practical assignment will make part of the credit together with a written test.
- Take your own notes in the lecture hall as they will be essential.



Requirements

- Recommended Textbooks:
 - Java: How to Program; Deitel and Deitel; ISBN 0-13-012507-5
 - C#:How To Program; Deitel and Deitel; ISBN 0-13-06222104
 - Object-Oriented Analysis and Design with Applications; Grady Booch; ISBN 0-8053-5340-2
- Java Compiler and Documentation:
 - Available at <http://java.sun.com> as Java 2 SDK 1.3 or 1.4 (J2SE)
 - Docs at <http://java.sun.com/j2se/1.4.1/docs/api/index.html>
 - Remember to add the bin directory to your path (in autoexec.bat for Windows)
 - Command prompt programming is recommended ('command' or 'cmd')
 - Yet you can use any IDE such as JBuilder and Forte
- C# Compiler and Documentation
 - Csc.exe is a free C# compiler; alternative Visual Studio .net
 - Docs at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/cpref_start.asp



OOP

Why Java/C# ?

- Java/C# is just another programming language but has several special feature:
 - Simple making it easy to learn and use
 - Reliable and Robust
 - Some of its features are formally proven
 - Has many built in features such as memory management and garbage collection
 - Most features are tested at run-time.
 - Platform independent
 - Never compiles to machine code but to byte-code/MSIL
 - Any platform using JVM/.Net (CLR) can run this byte code
 - Distributed
 - Secure
 - Interpreted making development easier
 - Multi-threaded
 - Dynamic making changes on the fly possible
 - Object Oriented



OOP

Why Object Oriented?

- Object Oriented Programming is an approach towards programming where we take a more natural view of the problem domain as compared to procedural programming.
- Main Features of OOP are:
 - Abstraction
 - Encapsulation
 - Modularity
 - Hierarchy
- Java/C# can still be used as a non-OOP language but it acts as an aid towards OOP. A more strict language is SmallTalk and a less strict language is C++.



Your First Program in Java

- Type the following using the edit command or notepad

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

- Save as HelloWorld.java
- Invoke Compiler : javac HelloWorld.java
- Invoke JVM : java HelloWorld
- Should see ‘Hello World!’ printed to screen.
- See that CLASSPATH is not set or set to current directory.
- doskey comes in handy.

OOP

Your First Program in C#

```
using System;
class Welcome1 {
    static void Main (String[] args) {
        Console.WriteLine
            ("Hello World!");
    }
}
```

- Compile using appropriate buttons, and then execute.



Sample Output

```
C:\Programs\Java>edit HelloWorld.java  
C:\Programs\Java>javac HelloWorld.java  
C:\Programs\Java>java HelloWorld  
Hello World!  
C:\Programs\Java>
```

OOP

Template for Programs

- The following template should be used until we cover what each keyword means.

```
import java.io.*;
public class ClassName {
    public static void main(String args[])
    {
        // your programming code
    }
}
```

```
using System;
class ClassName {
    static void Main(String args[])
    {
        // your programming code
    }
}
```

- ClassName can be changed to whatever you wish but always save your program in a file ClassName.java
- **Java/C# Is Case Sensitive**
- After compiling, a byte/MSIL file called ClassName.class/Filename.exe is created.

C#



Exercises

- Try to compile the HelloWorld program.
- Create programs under a different classname.
- Make your program print out your name.
- Make your program print out 3 copies of your name under each other.