SLIIT
Discover Your Future

Object Oriented Programming

Introduction to Java

SLIIT
Discover Your Future

Module Content

- Introduction to Java
- Object Oriented Concepts – Recap
- Collections and Generics
- Thread Implementation
- Design Patterns

SLIIT
Discover Your Future

Assessments

- Continuous Assessments – 50%
  - Mid Term Exam (MCQ) – 20% (Week 8)
  - Online Exam – 10% (Week 11)
  - Group Project Submission– 20% (Last Week)
- Final Examination (Online) – 50%

SLIIT
Discover Your Future

Lecturers

- Ms.Thilmi Anuththara Kuruppu – Lecturer in Charge (Malabe)

thilmi.k@sliit.lk

- Mr.Udara Samaratunga (Malabe)
- Ms.Janani Tharmaseelan (Malabe)
- Dr.Kalpani Manathunge(Metro)
- Ms.Shanika Abeyrathne(Kandy)

OOP – Learning and Support

- Lectures – 2 Hours
- Tutorial – 1 Hour (Sometimes embedded with Lectures)
- Lab – 2 Hours
- Homework – Typically ½ hour to 1 hour of work
- Quizzes – 5 minute Quizzes in each Lab similar to how OOC was conducted. Will be based on the Homework.
- Self Help Labs/Tutorials – Links to specific Help Labs/Tutorials will be provided so that you can cover material that you lack on your own.
- Help Desk – We will run a Help Desk from the 2nd Week onwards, you can get an appointment to get Help for OOP
- Video Lectures – Some of the Lectures will be recorded and made available in the course web.

How to get a good Grade

- Do the obvious things
  - Attend Lectures, Tutorials, Labs
  - Do your Assignments, Tutorials by yourself
  - Don't wait till the last minute to realize that you don't understand something. First try things on your own, go through the provided material, if you still can't get help from the OOP Team.
- The Final Exam is an Online Exam
  - Install Ecllipse Oxygen (Latest version) in your Home Computer.
  - Try out programs on your own.
  - Genuinely attempt the Homework that is given to you each week. This includes Tutorials given.
- Explore things on your own. Go through and try out material in the Internet about Java Programming.
- Work towards developing an impressive Group Project.

Group Project

- You need to be groups of 4 Members
- Same Group Members will be there for the Software Engineering Module.
- The Case Study will be given to you in the Software Engineering Module in Week 2.
- You will develop a Java Web Based Application. A recorded lecture introducing these topics will be made available from Week 3.
- Submission and Presentation are due in the last week of the Semester.

Asking Questions

- Talk to your Lecturer/Instructor Directly.
- Use the Courseweb Moodle Forum to ask Questions.
- Get an appointment for a Help Desk (Details from 2nd Week onwards)

SLIIT
Discover Your Future

Learning Outcomes

At the end of the Lecture students should be able to

- List the differences between C++ and Java
- List how a Java program is compiled and executed in a Computer.
- List Features that make Java unique
- Write, Compile and Execute simple Java programs
- Write a Java program including
  - Input / output commands
  - Variable
  - Sequence
  - Selection
  - Repetition

Java

- 1991 - James Gosling, Sun Microsystems, Inc.
- Originally a platform independent language for programming home appliances and was called “Oak” later renamed “Java” in 1995.

- Later (1994) used for World Wide Web applications (since byte code can be downloaded and run without compiling it)
- Eventually used as a general-purpose programming language (for the same reason as above plus it is object-oriented)
- Why the name “Java”? Java was then named “Java”, paying homage to the large amounts of coffee consumed by the team.
- Now owns by Oracle

- Full-fledged application programming language
- Additional capability as a Web programming language (currently the strength of its application base)
- A pure OO programming language
- NOT radical or especially new
- Adopts its looks from C++, and its behavior from Smalltalk
- Compiled to processor-neutral instruction set then interpreted on each supporting platform
- Extremely fast adoption rate! (due to WWW)

Buzzwords

Simple	Java has a concise, cohesive set of features that makes it easy to learn and use.
Secure	Java provides a secure means of creating Internet applications.
Portable	Java programs can execute in any environment for which there is a Java run-time system.
Object-oriented	Java embodies the modern, object-oriented programming philosophy.
Robust	Java encourages error-free programming by being strictly typed and performing run-time checks.
Multithreaded	Java provides integrated support for multithreaded programming.
Architecture-neutral	Java is not tied to a specific machine or operating system architecture.
Interpreted	Java supports cross-platform code through the use of Java bytecode.
High performance	The Java bytecode is highly optimized for speed of execution.
Distributed	Java was designed with the distributed environment of the Internet in mind.
Dynamic	Java programs carry with them substantial amounts of run-time type information that is used to verify and resolve accesses to objects at run time.

Terminology

- Class - A collection of data and methods that operate on that data.
- Method - A group of statements in a class that handle a task.
- Attribute - A property of an instance of a class.
- Interface - A skeleton class.
- Package - A group of logically related codes (classes & interfaces).

Terminology

- Bytecodes
  - A set of instructions that look like machine code, but are not specific to any processor.
- Virtual Machine
  - The environment in which Java runs. The JVM is responsible for executing the bytecodes and has responsibility for the fundamental capabilities of Java.

C++ vs Java

```cpp
// C++ Program
#include <iostream>

int main ( )
{
   std::cout << “Hello World !“ <<
                          std::endl;

   return 0;
}
```

Helloworld.cpp

```java
// Java Program

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

Helloworld.java

Output :

```
Hello World !
```

SLIIT
Discover Your Future

First Java Program

```
/* First Java Program
*/

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

Comments

```
// Java Program  :  prg_01.java
// Printing a String
```

- Comments provide information to the people who read the program
- Comments are removed by the preprocessor, therefore the compiler ignores them
- In Java, there are two types of comments
  - Single line comments //
  - Delimited comments /* */ for comments with more than one line.

Everything is Object Oriented

```
public class HelloWorld
```

- In Java is fully object oriented, even the simplest program needs to be written using a class.
- HelloWorld is the name of the class.
- A public class needs to be stored in a file name that matches the class name. i.e. The above code needs to be saved in a filename called HelloWorld.java

The `main` method

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

- Java programs begin executing at method `main`. The main method must be given as above.
- The main method is one of the methods in a class
- void methods do not return a value.

Output Statement

```
System.out.println("Hello World !");
```

- Instructs the computer to display the data within brackets to the screen.
- `"Hello World !"` : String / String Literal. What you need to display on screen
- `System.out.println()` moves the cursor to the next line after displaying the data

SLIIT
Discover Your Future

Exercise - 1

- Write a Java program to display your name and address in 3 lines.

SLIIT
Discover Your Future

Java Keywords

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

Java Development Environment

1. Install the Java JDK. The latest Java SE JDK is 9. We will use Java SE JDK 8 in this course.
2. Editing a program
   - Type a Java program (source code)
   - e.g :

     vi editor (Linux)

     notepad (Windows DOS prompt)

     IDE (Eclipse, IntelliJ)
   - Extension : .java

Java Development Environment cont…

3. Compiling a Java program

- The java compiler (javac) compiles java code to an intermediate language called Java Byte Code. This is a platform neutral low level language which can be translated to machine code. Compiled java programs are stored in files with the extension .class

4. Running a Java Program

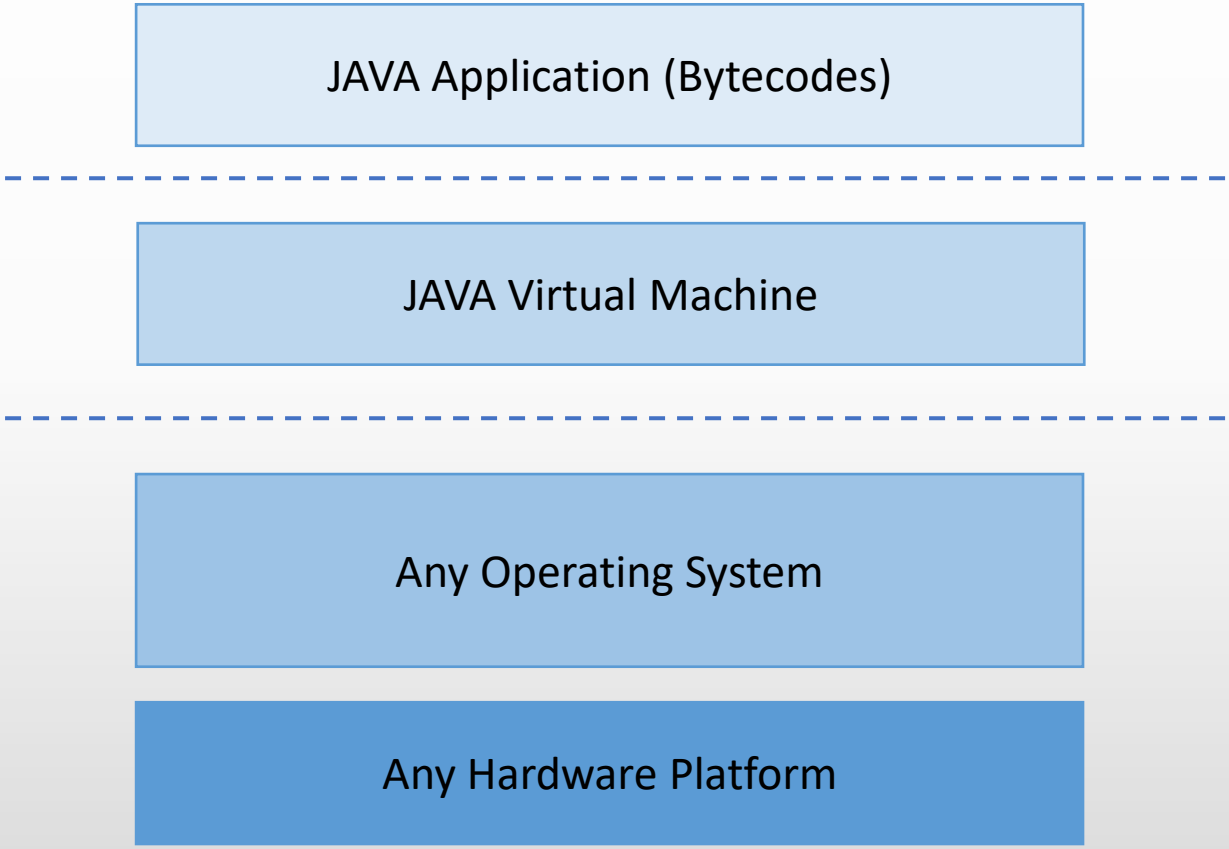
- The java interpreter is used to run a compiled program on your computer. Each platform has its own Java Virtual Machine which translates Java Bytecode to machine code.

5.

Integrated Development Environments (IDEs)

- Provides tools that support the software development process
- Includes, editors for writing & editing programs, debuggers for locating logical errors, design tools etc….
  - Eclipse
  - IntelliJ

Java Architecture



SLIIT
Discover Your Future

Platform Independence

Java source

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

Java compile

```
javac HelloWorld.java
```

Java bytecode

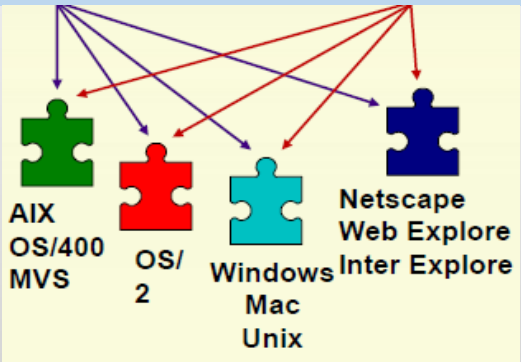
```
HelloWorld.class
-Platform Neutral
```

- **Write application once, runs on any**

- **Written in software - defined by Oracle**
  Assured through compatibility test suite

Virtual Machine - spec of microprocessor

Interpret **Translate**



Vendors Port VM to:

- *Operating Systems*
- *Enable in Browsers*

Java source compiled into intermediary bytecode

- *Application runs anywhere Java VM is ported*
- *Applet runs in any Java enabled browser*

Java's magic Byte Code

- Generated by Java compiler
  - Instead of generating machine language as most compilers do, the Java compiler generates byte code.
- Easily translated to machine language of various kinds of computers
- Executed by Java interpreter
- Invisible to programmer
  - You don't have to know anything about how byte code works to write a Java program.

Language Translation

A ***source program*** is the one that you write in the Java language and that always has a file extension of *.java*.

An ***object program*** is the binary byte-code program generated by the Java compiler, which always has a file extension of ***.class***.

The ***.class*** file generated by the Java compiler contains *bytecode* which is a low-level code similar to machine language, but generic and not specific to any particular CPU.

Java Virtual Machine

A given computer must have its own Java interpreter as part of a Java Virtual Machine, or JVM, to translate the generic bytecode into machine language for that CPU.

Virtual Machine (VM) interprets bytecodes into native machine language and runs it. Different VM exists for different computers, since bytecode does not correspond to a real machine.

Why Use Byte Code?

Disadvantages:

- Requires both compiler and interpreter
- Slower program execution

Advantages:

- Portability
  - Very important
  - Same program can run on computers of different types (useful with the Internet)
- Small in size
  - Linking at runtime
  - Easy to exchange over a network

Java Program Patterns

- Stand Alone Applications
  - Java programs which run in command console
  - Applications with GUI (awt/swing)

- Web-based Java Applications
  - Servlets
    - Programs that run inside request/response oriented servers
  - JavaServer Pages
    - An extension to the servlet architecture. Allow for the separation of the display of web content and the generation of that content

J…. confusion

- JVM (Java Virtual Machine)
  - JVM is a part of both the JDK and JRE that translates Java byte codes and executes them as native code on the client machine.
- JRE (Java Runtime Environment)
  - It is the environment provided for the java programs to get executed. It contains a JVM, class libraries, and other supporting files. It does not contain any development tools such as compiler, debugger, etc.
- JDK (Java Development Kit)
  - JDK contains tools needed to develop the Java programs (javac, java, javadoc, appletviewer,jdb, javap, rmic....),and a JRE to run the programs.
- Java SDK (Java software development kit)
  - SDK comprises a JDK and extra software, such as application servers, debuggers, and documentation.
- Java SE
  - Java Platform, Standard Edition (Java SE) lets you develop and deploy Java applications on desktops and servers (Same as SDK)
- J2SE, J2ME, J2EE
  - Any Java edition from 1.2 to 1.5

Exercise

What is the meaning of “write once, run anywhere”? Select the correct options:

1. Java code can be written by one team member and executed by other team members.
2. It is for marketing purposes only.
3. It enables Java programs to be compiled once and can be executed by any JVM without recompilation.
4. Old Java code doesn’t need recompilation when newer versions of JVMs are released.

Printing Values

- System.out.println() and System.out.print()

```
public class Print1 {

    public static void main(String args[]) {
        System.out.print("This is line 1");
        System.out.print(" still line 1");
        System.out.println(" lets move to the next line ");
        System.out.println("finally line 2");
    }
```

Print1.java

Printing Values

- System.out.println() and System.out.print()

```
int no = 50;
long population = 70000000;
double salary = 4500.34;
float rate = 34.5f;

System.out.println("no = " + no);
System.out.println("population = " + population);
System.out.println("salary = " + salary);
System.out.println("rate = " + rate);
```

Print2.java

Printing Values

- System.out.println() and System.out.print()

```
int no = 50;
long population = 70000000;
double salary = 4500.34;
float rate = 34.5f;

System.out.println("no = " + no + "\n"
        + "population = " + population + "\n"
        + "salary = " + salary + "\n"
        + "rate = " + rate);
```

Print3.java

Inputing Values from the Keyboard

- Using java.util.Scanner

```
3   import java.util.Scanner;
4
5   public class Input {
6       public static void main(String args[]) {
7           String name;
8           int age;
9           float salary;
10          Scanner myScanner  = new Scanner(System.in);
11          System.out.print("Enter your name : ");
12          name = myScanner.next();
13          System.out.print("Enter your age : ");
14          age = myScanner.nextInt();
15          System.out.print("Enter your salary : ");
16          salary = myScanner.nextFloat();
17
18          System.out.println("Name = " + name);
```

Input.java

Java vs C++ Language

- Java control structures are identical in syntax
  - selection – if, switch,
  - Repetition – while, do while, for
- Within a method the major difference is the print commands and the input commands.
- The basic data types integers and float are used in the same way. There is a separate data type in Java for string data called String.
- Calculations are also identical.
- There is a slight difference on how arrays are declared (i.e. similar to C++ dynamic arrays)

Use of Variables

- Same as in C++ (See Variables.cpp)

```
public class Variables {
    public static void main(String args[]) {
        int no = 50;
        long population = 70000000;
        double salary = 4500.34;
        float rate = 34.5f;

        System.out.println(no);
        System.out.println(population);
        System.out.println(salary);
        System.out.println(rate);
```

Variables.java

Calculations

- Same as in C++ (Calculations.cpp)

```
int no = 50;
long population = 70000000;
double salary = 4500.34;
float radius = 30.0f;

int remainder = no % 3;
double contribution = population * 100;
double area = 22.0/7*radius*radius;
```

Calculations.java

Selection - If

- Same as in C++

```
System.out.println("5. Kurunagala");
System.out.println("6. Jaffna");
System.out.println("0. Exit");

opt = 5;
System.out.print("Option : ");
System.out.println(opt);

if (opt == 1)
    System.out.println("Malabe Campus");
else if (opt == 2)
    System.out.println("Metro Campus");
else if (opt == 3)
    System.out.println("Matara Centre");
else if (opt == 4)
    System.out.println("Kandy Centre");
```

If.java

Selection - Switch

- Same as in C++

```
System.out.println("5. Kurunagala");
System.out.println("6. Jaffna");
System.out.println("0. Exit");

opt = -5;
System.out.print("Option : ");
System.out.println(opt);
switch (opt) {
    case 1 : System.out.println("Malabe Campus");
             break;
    case 2 : System.out.println("Metro Campus");
             break;
    case 3 : System.out.println("Matara Centre");
             break;
```

Switch.java

Repetition - while

- Same as in C++

```
int r = 1;
while (r < 100) {
    System.out.println(r);
    r++;
}
System.out.println();
r = 50;
while (r > 0) {
    System.out.print(r + " ");
    r -= 5;
}
```

While.java

Repetition - for

- Same as in C++

```
for (int r = 1; r<100; r++) {
    System.out.println(r);
}
System.out.println();
for (int r = 50; r > 0; r-=5) {
    System.out.print(r + " ");
}
```

For.java

Repetition – do while

- Same as in C++

```
int r = 1;
do {
    System.out.println(r);
    r++;
} while (r < 100);
System.out.println();
r = 50;
do {
    System.out.print(r + " ");
    r -= 5;
} while (r > 0);
```

DoWhile.java

Exercise - 2

- Write a java program to input the length and the width of a rectangle and calculate and print the perimeter.

SLIIT
Discover Your Future

Exercise - 3

- Write a program to input 3 integers and print the largest ad the smallest of the 3 numbers entered.

Exercise - 4

- Write a program to input 10 numbers from the keyboard and find how many odd numbers and how many even numbers were entered.