Day 1

Language

```
* C, C++, Java, C#, Python, GO, Ruby etc.
* Data types
* Syntax and Semantics
* Tokens
    1. Identifier
    2. Keyword
    3. Constant / Literals
    4. Operators
    5. Punctuators / Separators
* Built in features
* To implement B.L we should lanaguage.
* Types of application
    1. Console User Interface( CUI )
    2. Graphical User Interface( GUI )
    3. Library Application( .jar )
```

Technology

```
* ASP.NET, Java etc.
* It is used to develop application.
* Every language can be considered as Technology but every Technology can
not be considered as language.
```

Platform

```
* It provides environment in which we can run application.
* Types of platform
    1. Hardware based platform
        - All Operating Systems.
    2. Software Only Platform
        - Java, MS.NET Framework
```

Framework

```
* It is a library of reusable classes which is used to develop
application.
* RMI( Remote Method Invocation)
```

```
    - It is distributed application development Framework.
* AWT, Swing
    - It is GUI application development Framework.
* Collection Framework
    - It is library of Data structure and algorithms.
* Struts
    - It is MVC based web application development Framework
* Hibernate
    - It is automatic persistance Framework.
* JUNIT
    - It is a testing framework
```

Java language is both Technology as well as Platform.

Java History

```
* Core Java Vol - 1 ( Cay Horstman )
* It is developed in 1991 at Sun Micro Systems.
* It is invented By James Gosling and his Team ( Green Team).
* Initial Name of Java was "Oak". But due to name ambiguity it is renamed
to "Java". It is name of coffee.
* First version of java(1.0) was relased in 1996.
* Now it is product of Oracle.
* Current version of java is : Java 13
```

Java Platforms

```
1. Java SE
2. Java EE
3. Java ME
4. Java FX
```

Java SE

```
* Java Standard Edition.
* It is also called as core java.
* It is used to develop CUI, GUI, networking application, distributed
application, libraries etc.
* Java SE API's are sub set of Java EE API.
```

Java EE / JEE

```
* Java Enterprise Edition
* It is also called as Web Java/Enterprise Java / Advanced Java.
* It is used to develop web application and web services.
* Java EE API's are super set of Java SE and ME API's
```

Java ME

```
* Java Micro Edition
* It is used to develop application for consumer devices.
* e.g Mobile
* It is sub set of Java SE API.
```

Java Fx

```
* It is used to develop rich GUI application for internet then we should
use Java Fx.
```

Java SE Platform / Core Java

```
* Java Platform consist of two components
    1. Java Application Programming Interface( Java API ).
    2. Java Virtual Machine( JVM ).
* In java, interface/class/enum is generally refered as Java API.
* "rt.jar" is a java library file which contains implementation of Java
API.
* JVM is runtime environment of java which is used to execute any java
applicattion.
* It is also called as abstract computer.
* rt.jar and JVM are integral part of JRE( Java Runtime environment ).
* JRE is a software.
* If we want to deploy java application on client's machine then we must
install JRE on that machine.
```

SDK = Lang Tools + Documentation + Supporting Libs + Runtime environment.

JDK = Java Lang Tools(bin) + Java Docs (docs) + rt.jar + JVM

JDK = Java Lang Tools + Java Docs + JRE[rt.jar + JVM]

jdk 1.8 is a software which supports all the features specified in Java SE 8.

JDK Installation Directory Structure

1. bin
    - It contains java language development tools.
2. include
    - In context of java, C/C++ code is called native code.
    - JNI : Java Native Interface, it is java's framework used to access native code in java.
    - Using JNI, if we want to access native code in java then we need to use header files declared in include directory.
    - JNI : reference : "www.artima.com"
3. lib
    - It contains jar files required for third party tools
    - e.g IDE, Build tools
4. src (src.zip)
    - It contains source code of Java API.
5. jre
    - It contains JVM implementation and rt.jar
6. docs
    - It contains documentation of Java API
7. man
    - It contains documentation of java language tools.

Type

```
It is general term used in java to refer built in type as well as user
defined type.
```

Organization of types:

```
* All the types are organized in jar file.
* Jar File Contains
    1. Package(s)
    2. Menifest File ( Binary File )
* Package can contain
    1. Sub Package
    2. Interface
    3. Class
    4. Enum
    5. Error
    6. Exception
    7. Annotation
* Interface can contain
    1. Final Field / constant
    2. Abstract method
    3. Default method
    4. Static method
    5. Nested Interface
* Class Can contain
```

```
    1. Nested Type( interface /class/enum )
    2. Field
    3. Constructor
    4. Method
```

Java Terminology

```
*   C++ :   Java
* class  :   class
* Object   :    Instance
* Base class    :   Super class
* Derived class :   Sub class
* Access Specifier  :   Access Modifier
* Namespace     :   Package
* Data Member   :   Field
* Member Function   :   Method
* this pointer  :   this reference
* Concrete class : We can instantiate concrete class
* Abstract class : We can not instantiate abstract class
* Final class : We can not extend final class.
```

Naming/Coding Conventions

1. Camel Case Naming Convention
2. Pascal Case Naming Convention

Camel Case

```
* e.g
    1. main()
    2. parseInt()
    3. showInputDialog( )
* In this case, except first word, first character of each word must be in
upper case.
* In java, it is used for
    1. Field name
    2. Method name
    3. Method parameter and local variable
    4. Object reference.
```

Pascal Case

```
* e.g.
    1. System
```

```
    2. StringBuilder
    3. NullPointerException
    4. IndexOutOfBoundsException
* In this case, including first word, first character of each word must be
in upper case.
* In java, it is used for
    1. Type Name( Interface/class/Enum )
    2. File Name
```

Name of the package should be in lower case.

Name of final field and enum constant must be in upper case.

- src.zip : It constains source code of java API.
- rt.jar : It contains compiled code of java API
- java docs : It contains documentation of Java API.
- rt.jar file contains following "main packages":
    1. com
    2. java
    3. javax
    4. org
    5. sun
- java main package contains 14 "sub" packages
    1. applet
    2. awt
    3. beans
    4. io
    5. lang
    6. math
    7. net
    8. nio
    9. rmi
    10. security
    11. sql
    12. text
    13. time
    14. util
- If we want to group functionally equivalent / related types together then we should use package.
- java.lang package contains all the fundamental types of core java.

"Hello World!!" Application

```
Step 1 : vim Hello.java
Step 2 :
```

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

```
Step 3: javac Hello.java ( Press enter )
Step 4: java Program
* Java Compiler generates .class file. It contains "bytecode".
* Bytecode is object oriented assembly language code designed for JVM.
* Using "javap -c" option we can see bytecode.
* eg.   javap -c Program.class
* Executing bytecode is a job of JVM.
```

System

```
* It is a final class declared in java.lang package.
* Following are fields of System class
    1. System.in
    2. System.out
    3. System.err
```

Definition of System class

```
package java.lang;
public final class System
{
    //Fields
    public final static InputStream in;
    public final static PrintStream out;
    public final static PrintStream err;
    //Methods
    public static Console console(){   }
    public static void gc(){   }
    public static void exit(int status){   }
}
```

System.out

```
* out is object reference / reference of java.io.PrintStream class.
* out is declared as public static final field inside System class.
```

println() method

```
* print(), printf() and println() are non static methods of
java.io.PrintStream class.
```