***Java Technology & First Cup of Java***

***Certified Professional in Java Programming***

***Tutorial 1***

1. **What are the different types of Java solution available?**

* Desktop Application
* Mobile Applications
* Java Servlets – a class which responds to a network request. (Typically HTTP requests)

Responds by implementing methods like init() / doGet() / doPoast()

Servlets are server-side Java EE (Enterprise Edition) components that generate responses (typically HTML pages) to requests (typically HTTP requests) from clients.

* Java Applets – a java program that runs in a Web browser. (a web page displayed in a web browser) A JVM is required to view an applet.

1. **Java is not only a programming language, but it is a technology. Explain**

The Java programming language is a high-level language that can be characterized by all the following buzzwords:

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

1. **What is a Platform**

* A platform is the hardware or software environment in which a program runs. Some of

the most popular platforms are Microsoft Windows, Linux, and Mac OS.

* Most platforms can be described as a combination of the operating system and underlying

hardware.

* The Java platform differs from most other platforms in that it's a software-only platform

that runs on top of other hardware-based platforms.

The Java platform has two components:

* The Java Virtual Machine - JVM
* The Java Application Programming Interface (API)

1. **Explain the platform-independent feature of the Java technology**

The meaning of platform-independent is that the java compiled code (byte code) can run on all operating systems. (Required JVM)

1. **Java is a complied and interpreted language. Explain**

Java Source code first compiled into a binary code using java compiler and this byte code runs on the JVM (interpreter). Which means java source codes can run any JVM installed platform.(platform independent)

1. **Explain the following terms?**

* **Javac command** - The javac command is used to compile Java programs, it takes .java file as input and produces bytecode. Following is the syntax of this command.

javac sample.java

* **Java command** – The java command is used to interpreter the byte code produced by the java compiler. Following is the syntax of this command.

java sample

* **Byte code** - Java bytecode is the instruction set for the Java Virtual Machine.

As soon as a java program is compiled, java bytecode is generated. java bytecode is the machine code in the form of a .class file.

* **JDK** – is a set of tools for developing a java application.

Platforms depend (Windows/ MacOS/ Linux)

* **JRE** - a software layer that runs on top of a computer’s operating system software and provides the class libraries and other resources that a specific Java program needs to run. JRE include in JDK.
* **JVM** – this enables a computer to run Java programs and other programs that compiled into a java bytecode.

The JVM performs following operation:

* Loads code
* Verifies code
* Executes code
* Provides runtime environment
* **path variable** – this used to locate the JDK packages that used to convert java source code into machine code.

Tools like javac and java can not be used without giving the path variable.

1. **Why do you need to update the path environment variable when developing java Programs?**

Setting path variable gives conveniently access to the executables. Without setting the path variable, required to mention path variable every time manually when executing the source code.

1. **Is java a case sensitive language ?**

Yes

Hello and hello can have different meaning

1. **What do mean by a syntax error, Runtime error, Logical error?**

* Syntax Errors / Compile Time Error: – these errors are occurred in program compile time, and they prevent the code from running.
  + missing bracket
  + missing semicolon
  + access to private fields in other classes
  + class not found.

Misspelled variable names or method names.( Case sensitivity)

* + Incorrect format of selection statements or loops

for (i = 1, i <= 10; i++) { }

// Should have been for(i=1; i<=10; i++)

* + Missing Parenthesis

// { { } } ( )[ ( ) ] – the parentheses are valid

// { { { } – the parentheses are invalid

These are some typical reasons that happened to occur syntax errors. The compiler indicates the error line and sometimes guesses the error.

* Runtime Errors - Run time error occurs during the program execution time. It's detected by JVM while the program is running. When these types of errors occurred noting is printed in the console, because the errors occurred after the compile time.

These types of errors are caused to crash the program. Usually, these errors occurred when the program does not contain any syntax errors, but programmers/users ask to do the task that the processor unable to process during the compile time. The best thing to handle the error to put the try block inside the error code and catch the error inside the catch block.

Example :-

public class RunTimeError{

public static int variable = 1;

public static int answer = 1/0;

public static void main(String args[]){

System.out.println(answer);

}

}

Result (console) :-

Exception in thread "main" java.lang.ExceptionInInitializerError

Caused by: java.lang.ArithmeticException: / by zero

at runtimeerror.RuntimeError.<clinit>(RuntimeError.java:18)

C:\Users\Dilan\AppData\Local\NetBeans\Cache\8.2rc\executor-snippets\run.xml:53: Java returned: 1

BUILD FAILED (total time: 0 seconds)

(throws an ArithmeticException in run time. Reason to throws the error when integer number divided by 0 it cannot print the answer - infinity)

* Logical Errors / Semantic Errors – these errors occurred when the program compiles and executes without any errors but does not return the expected answers. These errors are caused by the programmer side. Incorrect ideas, incorrect meanings, incorrect concepts caused incorrect results (Logical errors).

1. **What is the purpose of the main method in Java applications?**

JVM use main method as a program execution start point. Without main method program will not compile

1. **What is keyword ?**

Reserved words for java programing, these words cannot use these as constant , variables or identifiers.

(Constant :- variable whose value cannot change once it has been assigned.to define a variable as a constant, just need to add the keyword “final” in front of the variable declaration.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| abstract | assert | boolean | break | byte |
| case | catch | char | class | Const |
| continue | default | enum | extends | final |
| finally | float | for | goto | if |
| implements | import | instanceof | int | interface |
| long | native | new | package | private |
| protected | public | return | shot | static |
| strictfp | super | switch | synchronized | This |

1. **What is an identifier ?**

Identifiers are used to refer required items from another place. (Ex:- class)

1. **Mention all the identifier naming rules and conventions**

* Should start with the A to Z or a to z , can use \_ or $ as first letter
* After first letter can use any character
* Cannot use java keywords
* Case sensitive
* Meaningful
* Should consider the string value