JAVA Course

Course brief

Java is an object-oriented language that enables learners to create real-world applications. Java technology based software works just about everywhere--from the smallest devices to super computers! Java technology components are not impacted by the kind of computer, phone, smart device or operating systems they are running on. The architecture-neutral nature of Java technology is important in a networked world where one cannot predict the kind of devices that partners, suppliers and employees use to connect to their organizations. While most advanced certifications focus the test on your knowledge of the API, this entry level one focuses on variables, class and interface definition, arrays, exception handling, encapsulation, polymorphism and flow control.

Syllabus

Subjects	Duration
JAVA	6 weeks

JAVA

Curriculum

Java Refresher

Java Language features (quick overview). Object oriented programming. Creating Java Program (source file declaration, compilation, execution). Class access modifiers. What is an interface? Abstract class? Local, static, final variables?

Learning Outcomes

- Understand how to build Java applications and high level overview of the Java language features.

Object Orientation

Object Orientation in more detail. Constructors. Data Encapsulation. Inheritance. Is-a, has-a. Polymorphism. Overriding/Overloading. Creating interfaces and their concrete classes. Static variables and methods. Coupling and cohesion.

Learning Outcomes

- Declare Interfaces. Declare, Initialize, and Use Class Members. Use Overloading and Overriding. Develop Constructors. Describe Encapsulation, Coupling, and Cohesion.Use Polymorphism. Relate Modifiers and Inheritance. Use Superclass Constructors and Overloaded Constructors. Use IS-A and HAS-A Relationships.

Stack and heap

Passing variables into methods. Array declaration, Construction and Initialization. Boxing and unboxing. Using wrapper classes. Garbage collection.

Learning Outcomes

- Use Class Members. Develop Wrapper Code & Autoboxing Code. Determine the Effects of Passing Variables into Methods. Recognize when Objects Become Eligible for Garbage Collection.

Java operators

Assignment, Relational, Instanceof, arithmetic, conditional and logical operators.

Learning Outcomes

- To learn and apply the different types of operators supported in Java like arithmetic, conditional, relational etc.

Flow control and exceptions

Flow control (if, switch, labeled statements, while, for, do etc). Exceptions and its related keywords. Handling exceptions. Exception Hierarchy, Assertions (enabling and disabling assertions).

Learning Outcomes

- Use if and switch Statements. Develop for, do, and while Loops. Use break and continue Statements. Develop Code with Assertions. Use try, catch, and finally Statements. State the Effects of Exceptions. Recognize Common Exceptions.

Strings and Files

StringBuilder and String Buffer. File navigation and I/O. Serialization. Dates, Numbers and Currency. Parsing Tokening and Formatting.

Learning Outcomes

- Using String, StringBuilder, and StringBuffer. File I/O using the java.io package. Serialization using the java.io

package. Working with Dates, Numbers, and Currencies. Using Regular Expressions.

Collections

Collections overview. Object class methods (equal, hashcode etc). Different types of collections and their usage. Generic types. Polymorphism and Generics etc.

Learning Outcomes

- Design Using Collections. Override equals() and hashCode(). Distinguish == and equals(). Use Generic Versions of Collections Including Set, List, and Map. Use Type Parameters, Write Generics methods. Use java.util to Sort and Search. Use Comparable and Comparator.

Inner classes

Inner classes, Method local inner classes, Anonymous Inner classes, Static nested classes.

Learning Outcomes

- Using Inner Classes. Method-Local Inner Classes. Anonymous Inner Classes. Static Nested Classes

Threads

Defining, Instantiating, and Starting Threads. Thread States and Transitions Synchronizing Threads. Interthread communication

Learning Outcomes

- Threading concept, Start New Threads, Recognize Thread States and Transitions. Use Object Locking to Avoid Concurrent Access. Write Code that Uses wait(), notify(), or notifyAll().

Java and javac commands

Using the Javac and Java commands. Static imports. How to create JAR files.

Learning Outcomes

- How to use Packages and Imports. Determine Runtime Behavior for Classes and Command-Lines. Use Classes in JAR Files Use Classpaths to Compile Code.

Instructors

Mr.Mohammed Younus Shariff

Java

Younus is an M-Tech in Computer Science from Jawaharlal Nehru Technological University Hyderabad.

Prof. Mohammed Younus Shariff has over 10 years of satisfying training experience. His style of teaching leaves the students asking for more. Many a student has benefitted from interaction with him.

He is an Asst.Professor with the Department of Computer Science at The Keshav Memorial Institute of Technology (KMIT) Prof. Shariff teaches Python, Data Science with python, HTML5 , CSS3 and Android Programming.