WHEN YOU'VE GOT THE SKILLS, IT'S ALL YOURS TO PLAY FOR.

INDEX

TRAINER'S PROFILE

Gain advanced skills in analytics with India's leading experts through Sensei Guild Academy – our intensive course curriculum and dynamic training faculty will surely make you industry-ready, while keeping pace with innovation.

At Sensei Guild Academy, our mission is to inspire, educate and empower aspiring students with state of the art data skills. With an expert team of consultants, we bring forth a considerable domain experience in the field of big data analytics. Be it substantial experience, training excellence or blended learning opportunity, our instructors leave no stone unturned to usher you into the fascinating world of data and navigate through properly.

What to Expect from our Training Faculty?

- All our trainers possess in-depth practical understanding on the subject they teach; they are true blue experts in their respective fields of study.
- They are well-versed with a plethora of e-learning platforms – as well as possess adequate knowledge and expertise to perform systems-based training programs.
- Most of our trainers have over 7-8 years of experience in the field of data analytics and has previously worked across multiple verticals, including multinational organizations, business consulting and financial management firms, analytics boutiques, IT establishments and FMCG industries.
- An adequate industry exposure has not only made our instructors proficient to combine the best aspects of instructor-led and on-demand training, but also capable enough to design student's curriculum and other training materials, instead of focusing only on the teaching aspect. The technical content they devise is simple, lucid and student-friendly.
- Positive attitude, robust approach and solid team player enthusiasm – are a few characteristics you'll find in our trainers.

ANDROID DEVELOPMENT

DURATION
30 HOURS

BATCH
WEEKEND / WEEKDAY

PRE REQUISITE: CORE JAVA

ANDROID (CORE ANDROID)

BASICS OF ANDROID

- What is Android
- History and Version
- Installing softwares
- Setup Eclipse
- Hello Android example
- Internal Details
- Dalvik VM
- Software Stack
- Android Core Building Blocks
- Android Emulator
- AndroidManifest.xml
- R.java file
- Hide Title Bar
- Screen Orientation

UI WIDGETS

- Working with Button
- Toast
- Custom Toast
- Button
- Toggle Button
- Switch Button
- Image Button
- CheckBox
- AlertDialog
- Spinner
- AutoCompleteTextView
- RatingBar
- DatePicker
- TimePicker
- ProgressBar
- Quick Contact Budge
- Analog Clock and Digital Clock
- Working with hardware Button
- File Download

ACTIVITY, INTENT & FRAGMENT

- Activity Lifecycle
- Activity Example
- Implicit Intent
- Explicit Intent
- Fragment Lifecycle
- Fragment Example
- Dynamic Fragment

ANDROID MENU

- Option Menu
- Context Menu
- Popup Menu

ADAPTOR

- Array Adaptor
- ArrayList Adaptor
- Base Adaptor

LAYOUT MANAGER

- Relative Layout
- Linear Layout
- Table Layout
- Grid Layout
- Frame Layout
- Scroll View Layout

VIEW

- GridView
- WebView
- ScrollView
- SearchView
- TabHost
- DynamicListViewz
- ExpandedListV

ANDROID DEVELOPMENT

DURATION
60 HOURS

BATCH
WEEKEND / WEEKDAY

PRE REQUISITE: CORE JAVA

ADVANCE + CORE ANDROID

ANDROID SERVICE
- Android Service
- Android Service API
- Android Started Service
- Android Bound Service
- Android Service Life Cycle
- Android Service Example

DATA STORAGE
- SQLite Database
- Shared Preferences
- Internal Storage
- External Storage

SPEECH API
- TextToSpeech API
- TextToSpeech Example
- Managing speed and pitch
- Speech To Text

DEVICE CONNECTIVITY
- Bluetooth Tutorial
- List Paired Devices
- Working with WiFi
- Working with Camera

JSON
- JSON Parsing

COTENT PROVIDER
- Content Proivider Fundamental
- Contact Content Provider
- Other Built-in Content Providers
- Creating Custom Content Provider
- Understanding Content URI
- ContentResolver
- Sharing Information from custom content provider

ANDROID NOTIFICATION
- Notification API
- Creating Notification Builder
- Setting Notification Properties
- Attaching Actions
- Issuing Notification
- NotificationCompat.Builder class
- Android Notification Examples

MULTIMEDIA
- Wallpapaer
- Live Wallpaper
- Multimedia API
- Playing Audio
- Creating Audio Player
- Playing Video
- Alarm Manager
- Gallery

TELEPHONY API
- Telephony Manager
- Get Call State
- Call State BroadcastReceiver
- Simple Caller Talker
- Making Phone Call
- Send SMS
- Send Email

LOCATION API
- Location API Fundamental
- Example of Android Location API
- Working with Google Maps

ANIMATION
- Android Animation API
- Android Drawable class
- Android Animation Example
- Android Rotate Animation
- Android Fade Animation
- Android Zoom Animation

ANDROID GRAPHICS
- Graphics API
- 2D Graphics
- Android.graphics.Canvas
- Android.graphics.Paint class

DEVICE CONNECTIVITY
- Bluetooth Tutorial
- List Paired Devices
- Working with WiFi
- Working with Camera

SENSOR
- Sensor API
- Motion Sensor
- Position Sensor
- Environmental Sensor
- Sensor Values
- SensorManager class
- Sensor Class
- SensorEvent class
- SensorEventListener interface
- Compass Acceslerometer and Orientation Sensors
- Sensor Examples

ANDROID GOOGLE MAP
- Customizing Map
- GoogleMap class
- Android Google Map Application

PROJECT DEVELOPMENT
- Application Development

PYTHON DJANGO

SYLLABUS

INTRODUCTION TO THE BASICS OF PYTHON

- Explaining Python and highlighting its importance
- Setting up Python environment and discussing flow control
- Running Python scripts and exploring Python editors and IDEs

SEQUENCE AND FILE OPERATIONS

- Defining reserve keywords and command-line arguments
- Describing flow control and sequencing
- Indexing and slicing
- Learning the xrange() function
- Working around dictionaries and sets
- Working with files

FUNCTIONS, SORTING, ERRORS & EXCEPTION, REGULAR EXPRESSIONS, & PACKAGES

- Explaining functions and various forms of function arguments
- Learning variable scope, function parameters, and lambda functions
- Sorting using Python
- Exception handling
- Package installation
- Regular expressions

PYTHON: AN OOP IMPLEMENTATION

- Using class, objects, and attributes
- Developing applications based on OOP
- Learning about classes, objects and how they function together
- Explaining OOPs concepts including inheritance, encapsulation, and polymorphism, among others

DEBUGGING AND DATABASES

- Debugging Python scripts using PDB & IDE
- Classifying errors and developing test units
- Implementing databases using SQLite
- Performing CRUD operations

INTRODUCTION OF DJANGO FRAMEWORK

- Explaining web framework and MVC pattern
- Creating a basic Django app and views
- Using HTTP request and URLCONF
- Creating a simple view using Django

TEMPLATES AND FORMS

- Explaining Django template system
- Loading template files, and rendering templates
- Creating forms and processing form data
- Creating form field normalization
- Creating forms that can accept personal data from users

MODELS AND DYNAMIC WEB-PAGES

- Defining database models and using model fields
- Using QuerySets for data retrieval
- Using jQuery and AJAX with Django for creating websites

SERIALIZATION

- Explaining the concepts of data serialization
- Using the REST framework
- Using serializers, deserializers, and model serializers
- Creating REST APIs

PARSING XML AND JSON WITH PYTHON

- Describing JSON and XML file formats
- Explaining XML-RPC
- Parsing objects to XML and JSON and back, respectively
- Parsing data stored in XML/JSON format to native Python type and reversing the same

JAVA WITH SPRING BOOT

DURATION
80 HOURS

BATCH
WEEKEND / WEEKDAY

PRE REQUISITE: ANY PROGRAMMING LANGUAGE

SYLLABUS

- Java Basics
- Define the scope of variables
- Define the structure of a Java class
- Create executable Java applications with a main method; run a Java program from the command line; produce console output
- Import other Java packages to make them accessible in your code
- Compare and contrast the features and components of Java such as: platform independence, object orientation, encapsulation, etc.
- Working With Java Data Types
- Declare and initialize variables (including casting of primitive data types)
- Differentiate between object reference variables and primitive variables
- Know how to read or write to object fields
- Develop code that uses wrapper classes such as Boolean, Double, and Integer
- Classes & Objects
  Explain an Object's Lifecycle (creation, "dereference by reassignment" and garbage collection)
- Operator
  Using Operators and Decision Constructs
  Use Java operators; use parentheses to override operator precedence
  Test equality between Strings and other objects using == and equals ()
  Create if and if/else and ternary constructs
  Use a switch statement
- Array
  Creating and Using Arrays
  Declare, instantiate, initialize and use a one-dimensional array
  Declare, instantiate, initialize and use multi-dimensional arrays
- Iteration Statements
  Using Loop Constructs
  Create and use while loops
  Create and use for loops including the enhanced for loop
  Create and use do/while loops
  Compare loop constructs
  Use break and continue
- Working with Methods and Encapsulation
- Create methods with arguments and return values; including overloaded methods
- Apply the static keyword to methods and fields
- Create and overload constructors; differentiate between default and user defined constructors
- Apply access modifiers
- Apply encapsulation principles to a class
- Determine the effect upon object references and primitive values when they are passed into methods that change the values
- Working with Inheritance
- Describe inheritance and its benefits
- Develop code that makes use of polymorphism; develop code that overrides methods; differentiate between the type of a reference and the type of an object
- Determine when casting is necessary
- Use super and this to access objects & constructors
- Use abstract classes and interfaces
- Handling Exceptions
- Differentiate among checked exceptions, unchecked exceptions, and Errors
- Create a try-catch block and determine how exceptions alter normal program flow
- Describe the advantages of Exception handling
- Create and invoke a method that throws an exception
- Recognize common exception classes (such as NullPointerException, ArithmeticException, ArrayIndexOutOfBoundsException, ClassCastException)
- Multithreading
- Inner Classes
- Enums
- Generics and Collections
- Create and use a generic class
- Create and use ArrayList, TreeSet, TreeMap, and ArrayDeque objects
- Use java.util.Comparator and java.lang.Comparable interfaces
- Collections Streams and Filters

JAVA WITH SPRING BOOT

SYLLABUS

- Iterate using forEach methods of Streams and List
- Describe Stream interface and Stream pipeline
- Filter a collection by using lambda expressions
- Use method references with Streams
- Java I/O Fundamentals
- Read and write data from the console
- Use BufferedReader, BufferedWriter, File, FileReader, FileWriter, FileInputStream, FileOutputStream, ObjectOutputStream, ObjectInputStream, and PrintWriter in the java.io package
- Java File I/O (NIO.2)
- Use Path interface to operate on file and directory paths
- Use Files class to check, read, delete, copy, move, manage metadata of a file or directory
- Building Database Applications with JDBC
- Describe the interfaces that make up the core of the JDBC API including the Driver, Connection, Statement, and ResultSet interfaces and their relationship to provider implementations
- Identify the components required to connect to a database using the DriverManager class including the JDBC URL
- Submit queries and read results from the database including creating statements, returning result sets, iterating through the results, and properly closing result sets, statements, and connections
- Lambda Built-in Functional Interfaces
- Use the built-in interfaces included in the java.util.function package such as Predicate, Consumer, Function, and Supplier
- Develop code that uses primitive versions of functional interfaces
- Develop code that uses binary versions of functional interfaces
- Develop code that uses the UnaryOperator interface
- Java Stream API
- Develop code to extract data from an object using peek() and map() methods including primitive versions of the map() method
- Search for data by using search methods of the Stream classes including findFirst, findAny, anyMatch, allMatch, noneMatch
- Develop code that uses the Optional class
- Develop code that uses Stream data methods and calculation methods
- Sort a collection using Stream API
- Save results to a collection using the collect method and group/partition data using the Collectors class
- Use flatMap() methods in the Stream API
- JEE basics
- JEE 2-tier, 3-tier, n-tier architecture
- JEE application components-EJB, Servlets & JSP
- JEE runtime environment-Web Server & Application Server
- Servlets

  Introduction
  Web Application basics
  Client-Server Architecture
  Servlet Basics
  HTTP Servlet
  Servlet Lifecycle
  Servlet Engine
  Redirect & Request Dispatcher
  Session Tracking
  URL Writing
  HTTP Authentication
  Form Based Authentication
  Connectivity with Database using JDBC
- JSP

  JSP basics
  JSP Engine(Conversion to Servlet)
  Expressions and Declarations
  Directives
  Includes and Forwards
  JSTL

JAVA WITH SPRING BOOT

SYLLABUS

- Introduction to MVC architecture
- ORM Basics Introduction to Hibernate
- Hibernate Configuration Hibernate Sessions
- Annotation Based Configuration
- Hibernate O/R Mappings
- Inheritence Mappings
- Hibernate Query Language
- Criteria Queries
- JPA
- Introduction to Web Services
- SOAP
- REST
- Spring - Introduction and Architecture
- Spring - IoC Containers
- Spring - Bean Definition
- Spring - Bean Scopes
- Spring - Bean Life Cycle
- Spring - Bean Post Processors
- Spring - Bean Definition Inheritance
- Spring - Dependency Injection
- Spring - Injecting Inner Beans
- Spring - Injecting Collection
- Spring - Beans Auto-Wiring
- Annotation Based Configuration
- Spring - Java Based Configuration
- Spring - Web MVC Framework
- Spring - AOP with Spring Framework
- Spring - Transaction Management
- Spring Data
- Spring Boot
- Building REST Web Services using Spring MVC and Spring Boot

iOS App Development with Swift

SYLLABUS

- Define key programming terms relevant to Swift and iOS programming.
- Describe the process of creating iOS apps.
- State the purpose of the Apple developer tools, such as Xcode, Instruments, debugger, analyzer, and iOS Simulator.
- Distinguish well-written code from poorly-written code.
- Recognize patterns and idioms present in the Cocoa Touch API and other Apple frameworks.
- Employ the Apple developer tools to create an iOS app.
- Demonstrate programming best practices in Swift.
- Examine and subdivide app functionality into properly designed components.
- Explain and summarize iOS API features including location, mapping, sensors, gestures, multimedia and user interface components.
- Plan, prepare and build an original iOS app, from concept to working program.

SELENIUM

SYLLABUS

SELENIUM INTRODUCTION

- Test Automation
- Automation Advantages
- Introduction to Selenium
- Brief history of Selenium Project
- Selenium's Tools Suite
- Supported Browsers and platforms
- Differences between Selenium and QTP

SELENIUM IDE

- Selenium IDE introduction
- Installing IDE
- IDE features(ICONs)
- Building Test Cases
- Running Test Cases
- Selenium commands-Selenese
- Verifying Page Elements
- Assertion Vs Verification
- Locating Elements (Element Locators)
- Pattern Matching
- The "AndWait" Command
- WaitFor command in AJAX
- Store commands and Selenese Paramenters
- Echo – Selenium Print command
- Alerts, Popups & Multiple windows
- Debugging
- Writing a Test Suite
- IDE Test on different browsers
- Troubleshooting
- Selenium Test Runner

REALTIME SELENIUM PROJECT

- What is ajax
- Group Project
- Implementation Project

AI & MACHINE LEARNING COURSE

DURATION
60 HOURS

BATCH
WEEKEND / WEEKDAY

SYLLABUS

- INTRODUCTION TO PYTHON
- ENVIRONMENT SET UP
- PYTHON CRASH COURSE
- INTRODUCTION TO DATA SCIENCE
- PYTHON FOR DATA ANALYSIS - NUMPY
- PYTHON FOR DATA ANALYSIS - PANDAS
- PYTHON FOR DATA VISUALIZATION - MATPLOTLIB
- PYTHON FOR DATA VISUALIZATION - SEABORN
- INTRODUCTION TO MACHINE LEARNING
- MACHINE LEARNING - EXPLORING THE MODEL
- REGRESSION ALGORITHMS
- CLASSIFICATION ALGORITHMS
- CLUSTERING ALGORITHMS
- ENSEMBLE MODELS
- ASSOCIATION MINING
- DIMENSIONALITY REDUCTION ALGORITHMS
- MODEL SELECTION
- DEVOPS FOR DATA SCIENCE
- INTRODUCTION TO MLOPS
- INDUSTRY USE CASES

MICROSOFT AZURE

SYLLABUS

GETTING STARTED WITH MICROSOFT AZURE

- What Is Cloud Computing?
- What Is Azure?
- Managing Azure
- Subscription Management and Billing
- Lab : Using the Microsoft Azure Portal

WEBSITES AND CLOUD SERVICES

- Create and Configure Websites
- Deploy and Monitor Websites
- Create and Deploy Cloud Services
- Lab : Websites and Cloud Services

VIRTUAL MACHINES

- Create and Configure Virtual Machines
- Configure Disks
- Lab : Create a Virtual Machine in Microsoft Azure
- Create a Virtual Machine from the Gallery
- Verify the functionality of the Virtual Machine, Attach a Data Disk

VIRTUAL NETWORKS

- Getting Started with Virtual Networks
- Creating a Virtual Network
- Implementing Point-to-Site Networks
- Lab : Create a Virtual Network

CLOUD STORAGE

- Understand Cloud Storage
- Create and Manage Storage
- Lab : Configure Azure Storage

MICROSOFT AZURE DATABASES

- Understand Relational Database Deployment Options
- Create and Connect to SQL Databases
- Lab : Create an SQL Database

AZURE ACTIVE DIRECTORY

- Manage Azure AD Objects
- Manage Authentication
- Lab : Create Users in Azure Active Directory

MICROSOFT AZURE MANAGEMENT TOOLS

- Azure PowerShell
- The Azure SDK and the Azure Cross-Platform Command-Line Interface
- Lab : Using Microsoft Azure Management Tools

AMAZON WEB SERVICES

SYLLABUS

GETTING STARTED WITH AMAZON CLOUD

- Creating accounts and analyzing the cost breakdown
- Evaluating Service Level Agreements (SLA)
- Console, command line tools and API

OVERVIEW OF THE ARCHITECTURE

- EC2
- S3
- EBS
- Beanstalk
- RDS
- VPC
- CloudFront
- SimpleDB
- SQS
- CloudWatch

ACHIEVING AGILITY WITH EC2

MANAGING THE EC2 INFRASTRUCTURE

- Browsing Amazon Machine Images (AMI)
- Specifying security groups and key pairs

PROVISIONING RESOURCES

- Evaluating Elastic Block Store (EBS) and instance store root devices
- Assigning elastic IP addresses
- Mapping instance types to computing needs

IMPLEMENTING DURABLE AND RELIABLE STORAGE

STORING DATA IN THE CLOUD

- Persisting off-instance storage with EBS volumes
- Creating backups with snapshots
- Achieving high durability with SimpleStorage Service
- Transmitting data in/out of the Amazon cloud

SIMPLIFYING THE DATABASE INFRASTRUCTURE

- Achieving high availability of nonrelational data with SimpleDB
- Effortlessly implement a relational database with Relational Database Service (RDS)

CREATING COST-E ECTIVE DISTRIBUTED SOLUTIONS

- Decoupling applications with Simple Queue Service
- Leveraging CloudFront for high-performance edge cache content delivery
- Delivering static and streaming content

ADAPTING EC2 TO YOUR BUSINESS NEEDS

CUSTOMIZING VIRTUAL MACHINES

- Modifying existing images
- Creating new images off of running instances
- Converting an instance store AMI to an EBS A

CREATING AN AWS CLOUD ARCHITECTURE

- Applying best practices for a cloud solution
- Selecting a cloud setup for di erent use case scenarios

HANDLING DYNAMIC RESOURCE REQUIREMENTS

MONITORING FROM INSIDE OR OUTSIDE OF THE CLOUD

- Visualizing utilization metrics with CloudWatch
- Setting alarms to send and receive notifications

TRANSPARENTLY SCALING TO MEET LOAD VARIATIONS

- Distributing incoming traffic with elastic load balancing
- Dynamically adding and removing instances with Auto Scaling
- Setting capacity thresholds

HOSTING APPLICATIONS WITH ELASTIC BEANSTALK

MANAGING APPLICATION ENVIRONMENTS

- Customizing and configuring platform stacks
- Provisioning application resources with CloudFormation

AWS SECURITY FEATURES

CONTROLLING ACCOUNT SECURITY

- Configuring access credentials
- Managing users with Identity Access Management (IAM)

LEVERAGING THE VIRTUAL PRIVATE CLOUD (VPC)

- Provisioning isolated AWS resources
- Bridging EC2 instances to your internal network with a VPN
- Launching EC2 instances on dedicated hardware

SENSEI
GUILD ACADEMY
Transform. Achieve. Inspire

Buy the course at
www.senseiacademy.org

Courses conducted by SRC Education

Write to us at
hello@ufolympiad.com

WhatsApp to us at
+971 55 684 5812