CODEIT INSTITUTE OF TECHNOLOGY

AT CIT WE JUST DON'T CODE. WE CHANGE LIVES

CIT Python Cloud Software Engineering

About CIT

CODEIT Institute of Technology (CIT) has the distinction of being the first and only Nationally Recognized U.S. Department of Labor Apprenticeship Program in the state of New Jersey Specializing in Software Engineering and only one (1) of two (2) entities in the state of New Jersey to be designated an AWS Amazon Academy Institution. CIT is also a Middle States Association of Colleges and Schools Candidate.

CIT provides live instructor-led interactive video conferencing training to those least likely to be involved at the highest levels of technology with the skill sets and knowledge to be able to participate in a technology-driven workforce and to become the world's future innovators while creating generational wealth for themselves and their families We specialize in Python Software Enginerring, AWS Cloud Architecture and Block Chain Programming.

About Course

The goal of this course is to teach software engineering concepts through the Python Cloud Engineering language. In this course, you will learn fundamentals and advanced Python and AWS Cloud methodologies. The course will conclude with a dissertation project which will be a final project of your choosing for potential employers to look at and certification exams in AWS and Python.

Don't be afraid! You don't have to be a math wiz in any capacity to learn code!

INSTITUTE OF TECHNOLOGY

What you will learn:

- Python programming language
- AWS Cloud Solutions
- Programmatic logical thinking

What is Python?

Python is an interpreted high-level programming language for general-purpose programming created by Guido Van Rossum and was first released in 1991.

There will be a weekly recap quiz's to reinforce new topics.

There will be a number of projects that will be assigned as the course moves on to reinforce topics.

Resources:

- http://github.com
- http://pythontutor.com
- http://python.org
- https://www.tutorialspoint.com/python/index.htm

PLEASE READ:

This course is a chance to make a serious career change (google python developer salary), however as with any programming course, the class alone will not be enough. If you really want to make this change, you need to practice! 30 minutes to an hour of review or even external learning will go a very long way and will make it easier for you towards the end of the course when your your dissertation is due. The more effort you put in, the better chances of you getting hired quickly!

INSTITUTE OF TECHNOLOGY

Structure of Class:

2 hrs-Teaching, 2 hrs-In class coding
1 weekly quiz + assignment

Grading	
Quizzes	10%
Assignments	10%
Midterm	20%
Final Exam	35%
Dissertation	25%

Module 1: Starting to code – Python and idle

- **Installation and setup**
- **Rationale (why python?)**
  - Libraries
  - Show code snippets
  - Companies using it
  - Use cases
- **Thinking like a programmer**
  - Primitive expressions
    - ints, strings
  - Combining expressions
    - arithmetic, concatenation
  - Abstracting expressions
    - variables, (pure) functions
- **Assignments**
  - Temperature converter/bmi calculator

Module 2: Simple Programs

- **Bools**
- **If statements**
- **Numbers**
  - Limitations of floats
  - Use cases for modulo and floordiv
  - Type conversion, rounding
  - Math library
  - Use cases for diff number formats (hex, octal, bin)
- **How to read error messages**
- **Reading documentation, googling**
- **Assignment**
  - Cost of international calls
  - Triangle classification

Module 3: Strings, lists, Dictionaries

- **String cookbook**
  - Escapes
  - Formatting
  - Indices & slicing
- **Lists**
  - Use cases
  - Comprehensions
  - Matrices?
  - Reference equality gotchas

- **Dictionaries**
  - Use cases
  - Hashable req

- **Common patterns**
  - `in` and `not in` operators
  - When to pick which data structure
  - Common methods & funcs
  - `len`, `sorted`
  - `[::-1]`

- **`for`-loops**
  - Range
  - The iterable abstraction

- **Assignments**
  - Custom `max`/how to represent a phonebook/password strength
  - Calculator/fizzbuzz

Module 4: OOP

- **Basics of OOP**
  - Motivation

- **Instances/objects**

- **Encapsulation**

- **Inheritance**

- **Polymorphism**

- **Assignments**

Module 5: Bringing it all together

- **Exception handling**
  - bugs vs expected errors
- **File IO**
- **Debugging and testing**
- **Introspection with `dir`**
- **Modules (random, datetime, os, sys)**
- **Lambdas**
- **Code style**
- **Assignments**
  - Parsing .csv
  - Using pathlib

Module 6: Development Environments

- **Upgrade from idl**e
  - Editor
    - pycharm, vscode
  - Terminal
  - Debugger, linter, language server protocol
- **Virtual environments**
- **Git**
  - And github
  - Project structure
- **Pip & what you can ge**t
  - Requests, django/flask, numpy, pandas, tensorflow
- **Basic terminal interaction**
- **Assignments**
  - Personalize your dev environment

Module 7: Automation

- **APIs**
  - Hiding secret keys in environment variables
- **Twitter bot project (tweepy)**
- **Pwnedpasswords checker**
  - Requests, hashlib
- **Assignments**
  - Use {twilio, smtp} to send an {sms, email}

Module 8: Web scraping

- **Scraping basics**
  - Use cases, ethics
  - Json overview
  - How to inspect sites
- **Beautifulsoup**
  - Overview
  - Selectors
- **Scraping hacker news project**
- **Assignments**
  - Find top movies on IMDB since 1980

Module 9: Web Fundamentals

- **How the web works**
- **Building a {blog, portfolio} website**
  - Django/flask
  - Html/css templates
  - Database integration

- **Assignment**
  Customize the page we build in class

Week 10: Data science fundamentals

- **Jupyter notebooks**
- **Kaggle datasets**
- **Data wrangling with pandas**
- **Visualization w/ seaborn & bokeh**
- **Brief introduction to machine learning**
  - Use cases
  - Scikit-learn
  - Simple predictions using iris dataset
- **Assignment**
  - Scrape a site for data to wrangle

Module 11: Algorithms

- **Recursion**
- **Search & sort algorithms & their complexities**
  - {linear, binary} search
  - {insertion, merge, quick}sort
- **Fibonacci impls**
- **Towers of hanoi, pascal's triangle**
- **Assignment**
  - Walk through an explanation of {quicksort?} and why it works

Module 12: Data structures

- **Recognizing use cases and complexities of:**
  - Arrays
  - Stacks
  - Queues
  - Hash tables
- **Knowing how to implement:**
  - {Singly, Doubly} linked list
  - Binary tree
  - Graphs
- **DFS and BFS algorithms**
- **Assignment**
  - Check if a string of parentheses "()))((()())()" is balanced or not (use a stack)

Module 13: DevOps

- **Cloud instances (AWS)**
- **Dockerizing projects**
  - Using docker-compose to spin up a database image
- **MySQL / PostgreSQL**
- **Deploying to a domain**
- **Kubernetes scaling**
- **Assignment**
  - Deploy a "hello world" app for production, with all that entails

Module 14: Dissertation Project

- **Entire week devoted to your dissertation project**

- **Requirements:**
  - Project must have backend/frontend and serve a purpose
  - Code copied from the internet with no referencing will get a 0
  - Entire projects copied from the internet get a 0
  - Any code used from another source must be referenced with a link to the website
  - Code that does not work will get a 0
  - The student is expected to understand all their code (even that which was copied) and be able to explain it
  - Your dissertation is due Friday before class
    Presentations 10-20 minutes on Friday

CODEIT INSTITUTE OF TECHNOLOGY

AT CIT WE JUST DON'T CODE. WE CHANGE LIVES

CODEIT AWS Master Class Course Program

INSTITUTE OF TECHNOLOGY

What is AWS?

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster..

About Course

The goal of this course is to teach expert skills within DevOps, Cloud Infrastructure and AWS. In this course, you will learn:

- Working with AWS Cloud Services
- Learn to deploy architectures onto the AWS Cloud
- Deploying a Nodejs application using DynamoDB
- Deep dive into EC2 Instances and EBS Volumes
- Understand Snapshots and AMIs along with EFS Volumes
- Configuring and working Elastic Load Balancers (ELBs)
- Lean how to auto scale policies
- Learn building DevOps pipelines and best practices while working with AWS
- CodeStar
- Switch IDE and using AWS Cloud9, and much more.

INSTITUTE OF TECHNOLOGY

Instructor:

Elijah Kizito
elijahknsubuga@gmail.com

What you will learn:

- Become AWS Cloud Practitioner Certified
- Understand the core products of AWS
- Learn AWS Architectures in -depth
- Deploy full networks on the cloud
- Describe how all of the AWS services work together
- Get deep hands-on knowledge of AWS tools and services
- Learn how to migrate on-premise systems to the AWS cloud
- Gain a deep understanding of Cloud architecture, devops, and AWS solutions

Resources:

- AWS Documentation: https://docs.aws.amazon.com
- General Resources: https://docs.aws.amazon.com/#general_resources
- Guides and API References: https://docs.aws.amazon.com/#user_guides
- Tutorials and Projects: https://docs.aws.amazon.com/#tutorials
- SDKs and Toolkits: https://docs.aws.amazon.com/#sdks

Structure of Class:

- 1 hrs-Teaching, 1 hrs-In class practice (Practicle)

Module 1

Introduction

- Introduction
- Getting certified in AWS Services
- What is cloud computing
- Creating an AWS account
- AWS Management Console
- EC2 Dashboard Experience

- Innovative AWS Services - 6min
- AWS G4 EC2 Instance - NVIDIA
- End User Services
- Amazon AppStream
- Amazon Worklink
- Setting up a Worklink Account - Pre-Reqs
- Worklink Proof of Concept part 1
- Worklink Proof of Concept part 2
- Worklink Proof of Concept part 3
- Testing out Worklink with your mobile phone
- Test your Knowledge: Take a short quiz
- Test your Knowledge
- AR/VR2
- Virtual Reality with Sumerian
- Creating a basic Sumerian scene
- Test your Knowledge: Take a short quiz
- Test your Knowledge

AWS Firecracker

- What is AWS Firecracker
- AWS Firecracker design and architecture
- What are Bare Metal Instances? Learn AWS Bare Metal EC2 Instance
- AWS Firecracker and RUST Language

Test your Knowledge: Take a short quiz

- Test your Knowledge

Module 2

AWS Cost Management

- How to create Budgets in AWS?
- Cost and Usage Reports
- Creating a Billing Alarm

- **Security, Identity and Compliance**
  - Identity and Access Management
  - Configuring IAM
  - AWS Cognito
  - AWS Certificate Manager
  - AWS GuardDuty

- **Test your Knowledge: Take a short quiz**
  - Test your Knowledge

- **Compute (EC2)**
  - EC2 Instance basics
  - Launching EC2 Instances
  - Working with AMI
  - Snapshots and Restorations
  - Deep dive into EC2 Instances
  - Elastic Load Balancer basics
  - Setting up Elastic Load Balancers
  - Autoscaling Basics
  - Configuring Autoscaling

- **Test Your Knowledge**
  - Test Your Knowledge

- **Storage**
  - All about S3 Buckets
  - Cross-Region replication of S3 Buckets
  - Working with EBS Volumes
  - Creating EBS volumes
  - Working with EFS Volumes
  - Mounting EFS Volumes to EC2 Instances
  - Test your Knowledge: Take a short quiz • 0min
  - Test your Knowledge

- **Test your Knowledge: Take a short quiz • 0min**
  - Test your Knowledge

Module 3

- **Databases**
  - Database basics
  - Relational Database Service
  - Creating a MySQL DB Instance
  - Amazon Aurora
  - Working with Aurora
  - DynomoDB
  - A hands-on look at DynamoDB
  - AWS Elasticache

- **Management and Governance**
  - AWS Management Tools
  - The Management Console
  - AWS CloudWatch
  - AWS CloudTrail
  - AWS Command Line Interface
  - AWS Trusted Advisor
  - AWS CloudMap
  - The Well-Architected Tool

- **Cloud Networking**
  - Virtual Private Cloud
  - VPC Security
  - VPC Components
  - Creating and configuring a VPC
  - AWS CloudFront
  - Creating a CloudFront Distribution
  - Elastic Load Balancers

- Creating ELBs
- AWS Route53
- Hosting a domain with Route53

- **Creating Cloud Architectures**

  - Cloud Migration Basics
  - Batch process architecture
  - Media and content architecture
  - High availability and fault tolerance
  - Disaster recovery
  - Advertising architecture
  - Gaming architecture
  - File optimization architecture
  - Wordpress hosting architecture

Module 4

- **AWS CloudFormation**
  - What is CloudFormation
  - Launching and modifying an EC2 Instance
  - What are templates
  - CloudFormation Designer
  - Creating a webserver part 1
  - Creating a webserver part 2
  - Stack and stack sets
  - Modifying a stack template
  - Creating changesets
  - Creating a new stackset
  - Best Practices
  - Clean-up

- **Test Your Knowledge**
  - Test Your Knowledge

- **Serverless Computing**
  - What is serverless
  - Serverless App Repository
  - Benefits and Drawbacks of serverless
  - Chatbot architecture
  - The WebApp Architecture
  - Image Processing Architecture
  - Creating a slack chatbot
  - Creating an image moderation chatbot
  - Creating a static website
  - Configuring user access with Cognito
  - Creating a serverless backend
  - Working with RESTful APIs
  - Creating a Rekognition collection
  - Creating step functions
  - Modifying and testing step functions

- **Basics of Elastic Beanstalk**
  - What is Elastic Beanstalk
  - The environments of Elastic Beanstalk
  - How Permissions work
  - Shared Responsibility Model and Platforms
  - Launching an Elastic Beanstalk Environment

- **DevOps21 lectures**
  - AWS Developer Tools
  - AWS Automation Tools
  - AWS ToolKit Eclipse IDE
  - Github and Fork Repository
  - Installing Eclipse IDE Neon
  - Java App Project Brief
  - AWS Codestar First Project Java Application
  - Install Important Consideration

- Working with Lambda API Gateway
- Importing a CodeStar Project in Eclipse
- Creating a branch
- Adding a team member
- Switch IDE and using AWS Cloud9
- Deploy Java App in EBS Volume
- Working with JIRA
- JIRA Create/Issue
- Deploying WordPress Blog Using AWS CloudFormation
- Creating a PHP Web Application
- Working With AWS CloudFormation Template
- Working Within Eclipse for DevOps Pipeline
- Difference Between CI/CD

Blockchain in AWS

- Blockchain in AWS
- AWS Blockchain templates
- Creating a VPC
- Creating a keypair and security group
- Creating roles and policies
- Deploying a Hyperledger network in AWS
- Exploring the network
- The CLI container
- Creating a sample chaincode
- Installing and instantiating chaincode
- Invoking transactions and running queries
- What is a blockchain
- Components of a blockchain
- Ethereum vs Hyperledger
- Introduction to Hyperledger
- Hyperledger Fabric overview
- What is an identity
- Membership Service Providers

- **Machine Learning in AWS**
  - AWS Machine Learning Service
  - AWS ML Pricing
  - First steps in building a model
  - Understanding AWS datasources
  - ML Training models
  - Importance of feature transformation
  - Evaluating models
  - Preparing data
  - Creating datasources and models
  - Serverless ML Inference with Lambda
  - AWS SageMaker
  - Setting up SageMaker
  - SageMaker and Machine Learning
  - Intro to linear learner
  - Preparing data for linear learner
  - Training data with linear learner
  - Creating a hyperparameter tuning job

- **AWS Rekognition15 lectures**
  - AWS Billing Quick Check
  - Download and Install Pycharm
  - Create a new user: programmatic access
  - Logging in through the command line
  - Install Boto3 - AWS
  - Create images directory in Pycharm
  - Uploading images
  - Python pprint
  - Creating helpers file
  - Fetching labels from images
  - Extracting just labels
  - Face detection with Python
  - Using S3 label detection
  - Compare faces using python
  - Limits of AWS Rekognition