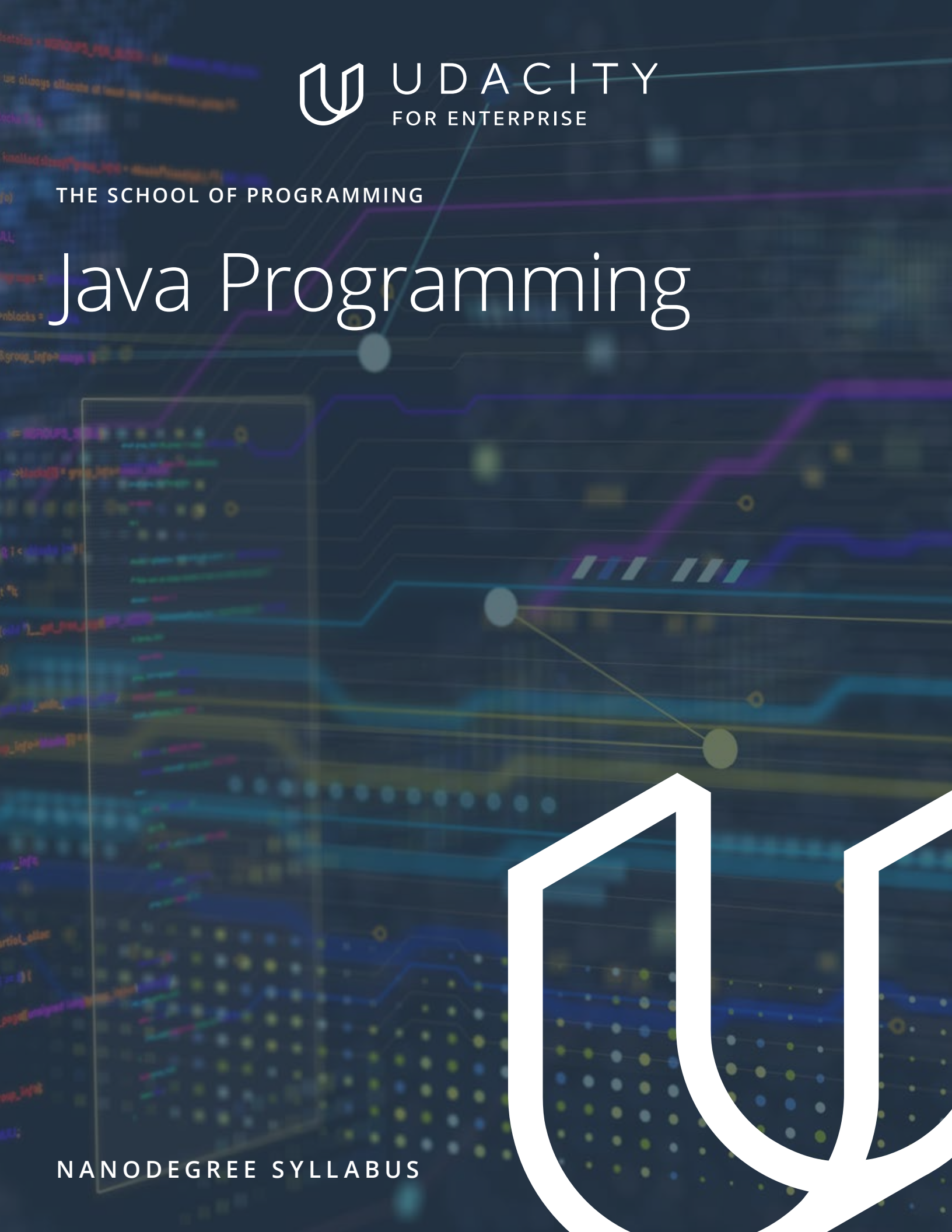

UDACITY
FOR ENTERPRISE
THE SCHOOL OF PROGRAMMING
Java Programming
NANODEGREE SYLLABUS

Overview

Java Programming Nanodegree Program

This Nanodegree program is intended to elevate your Java abilities and teach you the fundamental skills used by Java developers to design, code, test and deploy cutting-edge Java software. It is ideal for programming beginners with little or no Java experience who want to set themselves up for success as a professional Java developer and deploy functional Java-based applications of their own creation.

A graduate of this program will be able to:

- Apply basic features of Java programming language (data types, variables, conditional statements, loops, methods) to develop applications
- Use IntelliJ IDE to develop and troubleshoot Java applications
- Design and create classes and class methods in a Java application
- Apply interfaces, inheritance and polymorphism in Java applications
- Select and use Collections (lists, maps, sets, etc.) and basic features of functional programming (lambdas, streams, Optional) to organize and process data
- Apply advanced concepts (reflection, class loading, etc.) to implement advanced Java applications

Program Information

TIME
3 months
Study 10 hours/week

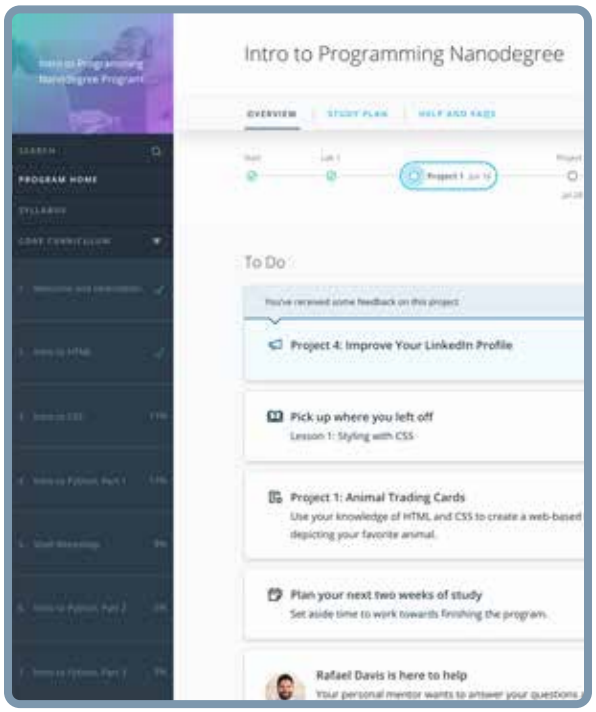
LEVEL
Foundational

PREREQUISITES
Basic programming skills

HARDWARE/SOFTWARE REQUIRED
Access to the internet and a 64-bit computer.

LEARN MORE ABOUT THIS NANODEGREE
Contact us at enterpriseNDs@udacity.com.

Our Classroom Experience



REAL-WORLD PROJECTS

Learners build new skills through industry-relevant projects and receive personalized feedback from our network of 900+ project reviewers. Our simple user interface makes it easy to submit projects as often as needed and receive unlimited feedback.

KNOWLEDGE

Answers to most questions can be found with Knowledge, our proprietary wiki. Learners can search questions asked by others and discover in real-time how to solve challenges.

LEARNER HUB

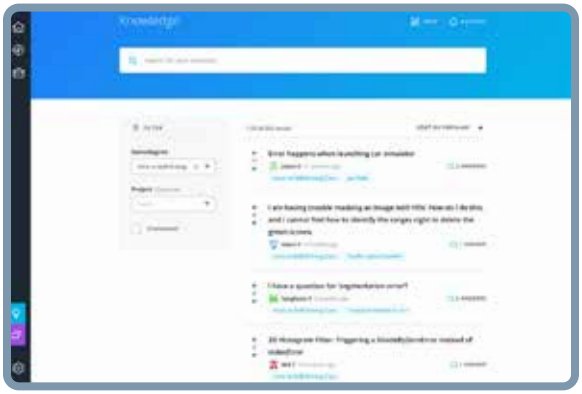
Learners leverage the power of community through a simple, yet powerful chat interface built within the classroom. Learner Hub connects learners with their technical mentor and fellow learners.

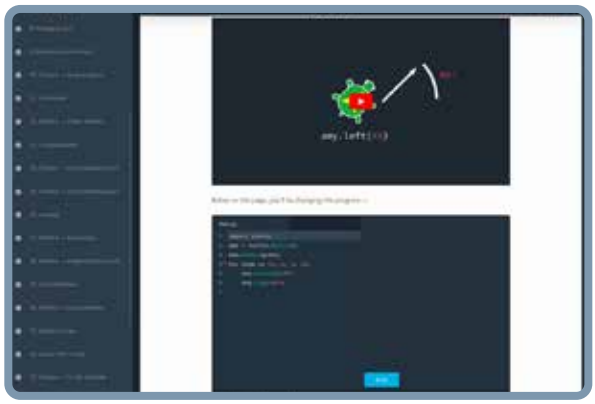
WORKSPACES

Learners can check the output and quality of their code by testing it on interactive workspaces that are integrated into the classroom.

QUIZZES

Understanding concepts learned during lessons is made simple with auto-graded quizzes. Learners can easily go back and brush up on concepts at anytime during the course.





CUSTOM STUDY PLANS

Mentors create a custom study plan tailored to learners' needs. This plan keeps track of progress toward learner goals.

PROGRESS TRACKER

Personalized milestone reminders help learners stay on track and focused as they work to complete their Nanodegree program.

Learn with the Best

Jeff Phillips

SENIOR SOFTWARE ENGINEER

During his more than 20 years of software development experience, Jeff has worked on embedded avionics flight controls systems for both Honeywell and Boeing. Jeff then moved into Java Cloud based SAAS applications. He has earned both a Bachelor's and Master's degree in Computer Science from Arizona State University.

Dustin Hellstern

SOFTWARE ENGINEER

Dustin is a software engineer with 15 years of Java experience, including over 7 years designing and building large-scale systems for one of the top companies in the tech industry. He is excited to share his knowledge with you in this program.

Alex Pritchard

SENIOR SOFTWARE ENGINEER

Alex is a Senior Software Engineer for CPA Global. He is excited to combine his background as a music educator with more than a decade of enterprise Java experience to help create this hands-on course about testing and deploying Java applications.

Course 1: Java Fundamentals

In order to be a great Java developer you must first learn the fundamentals of Java. This course introduces the fundamental concepts and practices of Java programming. These include basic Object-oriented Programming (OOP) concepts such as classes, encapsulation and inheritance, as well as core Java concepts such as collections, exceptions and commonly used Java types.

Project: Hotel Reservation Application

Students will be designing and implementing a hotel reservation application. The hotel reservation application will allow customers to find and book a hotel room. Rooms will contain the price along with the dates that the room is reserved. All room bookings will be associated with a single customer account, the customer account will include the customer name (first and last) and email. The application will allow customers to retrieve a list of the hotel's free rooms. In addition, the hotel reservation application will allow customers to find and book rooms based on availability and price.

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
JAVA PROGRAMMING BASICS	• Understand keywords and how they are used in Java • Be able to declare variables in Java • Describe the difference between primitive variables and objects • Apply casting to change the type of a variable into another type • Understand what truncation is and why it occurs • Create a method in Java • Describe the difference between Stack and Heap memory • Know the four different access modifier types • Create an array to store primitive variable types • Create each of the three different loops (While, do While and For) to iterate over an array • Create a Java program in IntelliJ • Describe the purpose of Javadoc
DEFINING CLASSES	• Explain what an object is and why we use them • Describe the differences between a class and an object • Know the parts and syntax for creating a class • Create different objects from a specific object type • Describe the purpose of Garbage Collection in Java
OBJECT-ORIENTED PROGRAMMING	• Describe what packages are in Java and why we use them • Apply Inheritance to share behavior and state between related classes • Apply polymorphism in Java to support decoupling • Describe the difference between an Interface and Abstract class

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
COMMON TYPE	• Apply throwing an Exception from a method when an issue is discovered • Create an Exception Handler to catch and handle an Exception • Use Enums to provide a predefined selection of constants • Use the Java Scanner class to read input from the console • Apply Java Data and Calendar class to store and manipulate day and time • Use the Java RegEx package to validate the input of a String • Use some of the advanced methods of the String class to create substring, search for text and investigate different characters of a string
GENERICS AND COLLECTIONS	• Describe the purpose of using Generics in Java • Create a generic method that processes different class types without casting • Apply generics to Collections to create stronger type data sets • Use data structures that implement the Collection interface • Use the Collections utility class to sort a list of Strings
ADVANCED JAVA TYPES	• Use Maps to store and retrieve data based on a key value • Use Sets to remove duplicate values from a List • Use a Queue to store values in a FIFO manner

Nanodegree Program Overview

Course 2: Advanced Java Programming Techniques

The best Java programs combine excellent software designs with modern language features. This course will teach you some lesser known features of Java, such as functional programming, I/O, serialization, and reflection. Strengthen your software design intuition by exploring the design ideas that underpin understandable, extensible and scalable software systems. Learn the basic concepts and techniques of concurrent programming in Java, so your programs can take advantage of modern, multi-core hardware.

Project: Legacy Web Crawler

Concurrency is a powerful tool to increase the performance of your Java programs. In this project, you'll use concurrent programming techniques to enhance a legacy web crawler so that it can take advantage of multi-core architectures. The crawler will read configuration from a JSON file, download and parse multiple HTML documents in parallel, and record popular web terms in an output file. You'll also build a method profiling tool to measure the efficiency of the crawler and prove the benefits of the parallel crawler.

LESSON TITLE	LEARNING OUTCOMES
FUNCTIONAL PROGRAMMING IN JAVA	• Understand the differences between imperative and functional programming • Use functional interfaces, lambdas, and method references to simplify and improve your Java code • Apply Java's Stream API to perform quantitative analysis over large sets of data
WORKING WITH FILES AND I/O	• Understand the differences between program memory and persistent storage • Know the different uses of binary and encoded data and understand when to use each • Use the Java Files API to read and write files • Recognize different kinds of resource leaks and apply modern Java techniques to prevent them • Serialize and deserialize between Java objects and common formats like JSON, XML and binary

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
DESIGN PATTERNS	• Identify the principles of good software design and build strong software design intuition • Understand what design patterns are and why they are useful • Use Creational, Behavioral and Structural design patterns to write flexible and easy-to-understand code • Apply Dependency Injection to simplify object creation and promote testable designs
REFLECTION	• Use reflection and annotations to introspect and add dynamic capabilities to your programs • Implement Java interfaces at runtime using dynamic proxies • Understand the fundamentals of Aspect Oriented Programming • Use class loaders to customize how Java loads byte code
INTRODUCTION TO CONCURRENT PROGRAMMING	• Recognize when concurrency can help improve the performance of Java programs • Use threads, thread pools and parallel streams to achieve parallelism in Java • Apply Java synchronization tools to correctly share state between threads in a parallel program

Nanodegree Program Overview

Course 3: Java Application Deployment

This course introduces Java ecosystem topics that are necessary to develop production-ready applications. It starts by covering the construction and makeup of Java program artifacts. You will learn how Java program code is compiled, packaged and executed. Next, you'll learn how to use Maven to automate and customize the build process, as well as manage external project dependencies. This course also covers the topic of Modules, introduced in Java 9. In addition to build topics, this course will also teach you to use the tools of JUnit 5 to write unit tests and evaluate code coverage. To expand our testing capabilities for complex applications, the Mockito library and test doubles will be covered as well.

Project: UdaSecurity

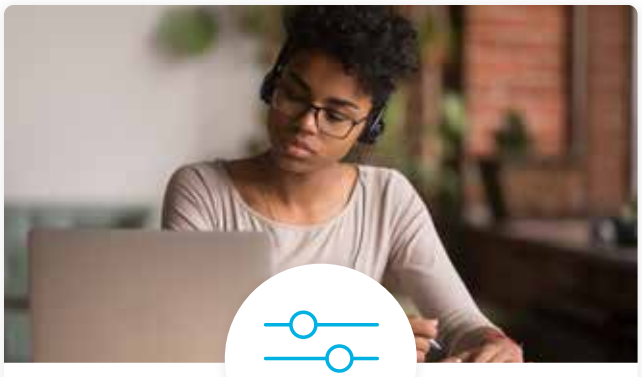
To practice all the skills covered in this course, you'll start with an existing project that needs help. The UdaSecurity program is a basic GUI application that allows users to perform various tasks related to managing their home security system. In order to prepare to scale the software, it's going to need some revisions. You'll need to refactor the program into a multi-module Maven project and you'll also be writing unit tests to verify that it actually does what it claims to do. You'll be using the JUnit 5 and Mockito libraries we cover in this course to write a full unit test suite for the project.

LESSON TITLE	LEARNING OUTCOMES
RUNNING JAVA APPLICATIONS	• Learn to use the tools of the JDK to compile, package and run Java applications • Recognize and evaluate bytecode • Use the JShell application to execute arbitrary Java code
DEPENDENCY MANAGEMENT WITH MAVEN	• Read and write well-formed XML, including Maven pom.xml files • Use Maven to build Java projects and include external dependencies • Use Maven plugins to modify the build process and perform additional tasks such as generating API documentation and executing static code analysis
JAVA MODULES	• Identify and create Java 9 modules using the Module Descriptor class • Convert existing projects into module format • Use the JLink tool to create a custom JRE for a specific Java module

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
UNIT TESTING WITH JAVA	• Explain the benefits of automated testing and identify the steps in the unit testing lifecycle • Write unit tests utilizing the features of JUnit 5 to cover all program requirements • Run unit tests automatically with Maven • Use tools in IntelliJ to identify any missing code coverage from your unit tests
MOCKING AND INTEGRATION TESTING	• Identify and create test doubles to isolate testing requirements from their dependencies • Use the Mockito library to create and modify the behavior of test doubles • Understand the roles of Unit Testing, Integration Testing, and Functional testing in the automated testing process • Create mock endpoints using Wiremock to isolate testing requirements from external API calls

Our Nanodegree Programs Include:



Pre-Assessments

Our in-depth workforce assessments identify your team's current level of knowledge in key areas. Results are used to generate custom learning paths designed to equip your workforce with the most applicable skill sets.

Dashboard & Progress Reports

Our interactive dashboard (enterprise management console) allows administrators to manage employee onboarding, track course progress, perform bulk enrollments and more.

Industry Validation & Reviews

Learners' progress and subject knowledge is tested and validated by industry experts and leaders from our advisory board. These in-depth reviews ensure your teams have achieved competency.

Real World Hands-on Projects

Through a series of rigorous, real-world projects, your employees learn and apply new techniques, analyze results, and produce actionable insights. Project portfolios demonstrate learners' growing proficiency and subject mastery.

Our Review Process

Real-life Reviewers for Real-life Projects

Real-world projects are at the core of our Nanodegree programs because hands-on learning is the best way to master a new skill. Receiving relevant feedback from an industry expert is a critical part of that learning process, and infinitely more useful than that from peers or automated grading systems. Udacity has a network of over 900 experienced project reviewers who provide personalized and timely feedback to help all learners succeed.

Vaibhav
UDACITY LEARNER

"I never felt overwhelmed while pursuing the Nanodegree program due to the valuable support of the reviewers, and now I am more confident in converting my ideas to reality."

now at
CODING VISIONS INFOTECH

All Learners Benefit From:

- Line-by-line feedback for coding projects
- Industry tips and best practices
- Advice on additional resources to research
- Unlimited submissions and feedback loops

How it Works

Real-world projects are integrated within the classroom experience, making for a seamless review process flow.

- Go through the lessons and work on the projects that follow
- Get help from your technical mentor, if needed
- Submit your project work
- Receive personalized feedback from the reviewer
- If the submission is not satisfactory, resubmit your project
- Continue submitting and receiving feedback from the reviewer until you successfully complete your project

About our Project Reviewers

Our expert project reviewers are evaluated against the highest standards and graded based on learners' progress. **Here's how they measure up to ensure your success.**

900+
Expert Project Reviewers
Are hand-picked to provide detailed feedback on your project submissions.

1.8M
Projects Reviewed
Our reviewers have extensive experience in guiding learners through their course projects.

3
Hours Average Turnaround
You can resubmit your project on the same day for additional feedback.

4.85 /5
Average Reviewer Rating
Our learners love the quality of the feedback they receive from our experienced reviewers.

UDACITY
FOR ENTERPRISE

2440 W El Camino Real, #101
Mountain View, CA 94040, USA - HQ

For more information visit: www.udacity.com/enterprise