



SOFTWARE DEVELOPER IN TEST PRACTICUM

PROGRAM OUTLINE

Full-Time Program
840 Hours of Classroom Instruction
210 Hours of Practicum
1050 Total Hours in Duration
70 Weeks in Total (Including Breaks)



PROGRAM DESCRIPTION

In this instructor led multidisciplinary program, students will learn the knowledge and skills required to become a junior/intermediate level Software Developer in Test within modern software development organizations.

The program starts with training in software testing and quality assurance methodologies and best practices as a precursor to introducing more technical automation testing and infrastructure concepts. Students will learn a variety of technical skill including Linux administration, virtualization and cloud computing, object-oriented programming, continuous integration and deployment strategies, performance-based testing, API testing, application deployment with containers and test automation with Selenium WebDriver.

This program's central focus will be developing students' programming ability for the purpose of creating a test automation framework. DevOps and infrastructure concepts, methodologies and tools will be introduced within the program to enhance the aspirant student's expertise of related technologies within the Software Development Lifecycle as a means of producing a highly skilled Software Developer in Test.

CAREER OCCUPATION PROGRAM

NOCs: 2173

This program prepares students for the following career occupations:

Software Testing Engineer, Application Architect, Computer Software Engineer, Software Design Engineer, Software Design Verification Engineer, Software Designer, Software System Integration Engineer, Software Technical Architect

ADMISSION REQUIREMENTS

Admission requirements may not be waived by either the student nor the Canadian College of Technology and Business (CCTB).

No funding may be disbursed to the student or received by CCTB until all admission requirements are met.

A payment plan can be offered to a student if student loan is not an available option.

Requirements:

- Good command of English language (See [Language Proficiency Policy](#))
- High school diploma or equivalent from an approved government institution of applicant's home country, or applicant is minimum 19 years of age
- Students are required to have and use their own personal computer in class



LEARNING OBJECTIVES

After completing this program, students will be able to:

- Identify all the organizational business and their respective roles and responsibilities
- Understand how each specific business unit contributes to the successful completion of the overall project as well as interoperates with the other units within the organization
- Understanding of different technologies and toolsets used by different roles within the organization
- Plan projects utilizing different SDLC models including Waterfall and Agile approach frameworks
- Analysis, interpretation, and creation of business requirements
- Translation of business requirements into strategic project objectives
- Be able to track issues and create defect reports in Application Lifecycle Management System
- Comprehend and apply the communication and defect lifecycle
- Describe frontend and backend development related concepts, tools and technologies including environment architecture, protocols, languages like Java and XML to the extent needed by Data Analyst
- Explain the underlying system architecture of Linux/Unix based systems
- Demonstrate proficiency in administering systems on a command line interface
- Utilize administrative commands on an advanced level
- Utilize package management, system services, text stream parsers, pseudo filesystem and system information commands
- Create, configure, and mount different filesystems and devices
- Demonstrate the ability perform networking configuration and administration on the command line interface
- Implement and configure Linux/Unix based graphical systems
- Implement, configure, and administer common Linux/Unix system services including SQL servers, printing systems and Web Servers
- Secure and audit systems
- Create and run bash shell scripts and system jobs for task management
- Understand the concept of data normalization and functional dependencies
- Understand database design methodologies and entity relationship modelling
- Use UML for diagramming notation
- Utilize SQL CRUD operations
- Demonstrate the knowledge and use of Data Definition Language (DDL) and Data Manipulation Language (DML)
- Understand database security, transaction management, triggers and stored procedures, replication, data warehousing and OLAP on a basic level
- Understand the architecture of virtualized and cloud-based systems and their implementation compared to traditional system infrastructure
- Deploy basic cloud-based infrastructure



- Demonstrate administrative abilities using the AWS Identity and Access Management Systems
- Utilize terminal client software to connect to and administer virtual machine instances
- Deploy applications using AWS services
- Design complex cloud-based infrastructure
- Utilize common AWS services to build highly available, scalable, robust, and secure system infrastructure
- Build properly configured and secure virtualized network environments
- Understand advanced AWS managed services and their utilization
- Design, build and deploy database solutions
- Deploy software applications using AWS deployment solutions
- Understand Java data types and language syntax
- Setup and manage a development environment
- Demonstrate an ability to use object-oriented programming with the usage of encapsulation, polymorphism, abstraction, and inheritance
- Properly utilize classes, methods, constructors, abstract classes, and interfaces
- Use Java collections and generics
- Debug code
- Develop unit tests using JUnit
- Perform Java input and output operations
- Utilize regular expressions
- Understand and utilize concurrency
- Implement SOLID design principles in their programming
- Use creational design patterns: builder, prototype, factories, singleton
- Use structural design patterns: adapter, bridge, composite, decorator, façade, flyweight, and proxy
- Use behavioural design patterns: chain of responsibility, command, interpreter, iterator, mediator, memento, null object, observer, state, strategy, template method, visitor
- Implement basic sorting and searching algorithms
- Implement basic data structures
- Understand and implement linked lists, binary trees, and hash tables
- Utilize browser development tools for element inspection and selection
- Run Selenium tests on multiple browsers as well as in headless mode
- Use CSS selectors and create dynamic XPath expressions
- Create End-to-End testing scripts with Selenium
- Use built-in methods and properties to create dynamic scripts
- Utilize waits, synchronization, and exception handling
- Perform JavaScript execution within Selenium scripts
- Work with advanced features such as iframes, window switching, keypress events and dynamic actions
- Use build management tools to setup an automation project
- Use TestNG for validation of test



- Utilize the Page Object Model
- Publish test results using reporting libraries
- Utilize logging within Selenium tests
- Perform automated database validation testing
- Build an Automation Testing Framework
- Demonstrate understanding of Web Application Architecture
- Demonstrate understanding of SOAP and REST APIs and the HTTP Protocol Architecture
- Read and write XML and JSON
- Perform API Testing with Postman
- Understand the REST Assured API Core Testing Library in Java
- Build an API Testing Framework
- Run basic performance test on JMeter using assertions, listeners, and recordings
- Utilize CLI, FTP, databases, web services API, dashboard reporting, plugin manager, reading CSV, function and variables, upload, download with JMeter
- Utilize parameterization, schedules and sequential execution, correlations, templates, debugging, configurations, distributed tests
- Install LoadRunner and understand its underlying architecture
- Design, setup and execute tests on LoadRunner
- Understand and deploy infrastructure as code (IaC)
- Utilize Ansible for automated server configuration and management
- Use Terraform for resource provisioning
- Create a monitoring system using Nagios
- Deploy Jenkins on AWS EC2 instances
- Build, configure, manage, and maintain a Jenkins server
- Build a continuous integration and delivery pipeline
- Create, deploy, and manage Docker containers
- Utilize Docker swarms and Kubernetes
- Integrate Jenkins and Docker for rapid application deployment

PROGRAM EVALUATION METHODS AND COMPLETION REQUIREMENTS

CCTB evaluates students using a variety of methods including projects, assignments, presentations, assessments, quizzes, and exams. Students will be given a performance evaluation before 30% of the hours of instruction of the program are completed. This evaluation will address any academic concerns that the college may have regarding student performance and/or learning outcomes. This evaluation will also ensure the student comprehensively understands the grading system, and what actions they can take moving forward to achieve or maintain a higher grade.

To complete the program, students will be required to achieve a minimum grade of 65% in each course, as well as complete the co-op/ practicum component of their program.

The co-op/practicum component of the program includes a performance-based evaluation conducted by the placement host and an analysis report created by the student relating to their work experience that must be submitted to the faculty.



Additionally, to successfully complete the program, students must maintain a minimum attendance rate of 75%.

If a course is failed, the student must re-take the course within the next available cohort. The course re-take fee is \$1100.

Please reference the CCTB [Dispute Resolution and Grade Appeal Policy](#)

HOMEWORK HOURS

A minimum of 2.5 - 3 hours of homework between lectures is to be expected.

DELIVERY METHODS

- Combined delivery (both in-class and distance)

REQUIRED PROGRAM MATERIALS

Resources in the form of custom learning materials will be provided by CCTB.

Software tools and user licenses will be provided by CCTB.

Instructors will provide students with additional educational resources that will be specific to the subject matter of each course. These resources will be used in conjunction with the class lectures.

These resources and learning materials will be made available online via the CCTB Canvas learning management system. Students are required to login to gain access to the e-materials.

Students must have and use their own personal computer in class.

Additional Recommended Learning Materials (not required):

- Nemeth, Snyder, Hein, Whaley (2010). Unix and Linux System Administration Handbook, 4th Edition. Prentice Hall.
- Tracy, R. (2015). CompTIA Linux+/LPIC-1 Certification All-in-One Exam Guide, Second Edition (Exams LX0-103 & LX0-104/101-400 & 102-400). McGraw-Hill Education.
- Sierra, Bates (2005). Head First Java: A Brain-Friendly Guide, Second Edition. O'Reilly Media.
- Schildt, H. (2014). Java: A Beginner's Guide, Sixth Edition. McGraw-Hill Education.
- Freeman, Robson (2004). Head First Design Patterns: A Brain-Friendly Guide, First Edition. O'Reilly Media.
- Lafore, R. (2002). Data Structures and Algorithms in Java, Second Edition. Sams Publishing.



- Geerling, Jeff. (2015). Ansible for DevOps: Server and configuration management for humans, 1st Edition. Midwestern Mac, LLC.
- Brikman, Yevgeniy. (2017). Terraform: Up and Running: Writing Infrastructure as Code 1st Edition. O'Reilly Media
- Laster, B. (2018). Jenkins 2: Up and Running: Evolve Your Deployment Pipeline for Next Generation Automation, First Edition. O'Reilly Media.
- Waud, E. (2018). Docker Quick Start Guide: Learn Docker Like a Boss, and Finally Own Your Applications. Packt Publishing

PROGRAM ORGANIZATION

1. Information Systems Management	60 HRS
2. Introduction to Linux Systems Administration	60 HRS
3. Relational Database Systems and SQL	60 HRS
4. Cloud Solution Architecture Level 1	60 HRS
5. Cloud Solution Architecture Level 2	60 HRS
6. Object Oriented Programming Level 1	60 HRS
7. Object Oriented Programming Level 2	60 HRS
8. Design Patterns, Data Structures and Algorithms	60 HRS
9. Software Test Automation Level 1	60 HRS
10. Software Test Automation Level 2	60 HRS
11. Introduction to Web Services and API Testing	60 HRS
12. Introduction to Performance Testing	60 HRS
13. DevOps Infrastructure Level 1	60 HRS
14. DevOps Infrastructure Level 2	60 HRS
Practicum Placement	210 HRS
Total Duration	1050 HRS