Model Curriculum

Computer Applications

SECTOR: IT-ITeS

SUB-SECTOR: Quality Assurance and Engineering

Occupation: Test Engineer-Product Development

REF ID: SSC/Q7001

NSQF Level : 4

QP Code: SSC/Q7001

Certificate

Table of Contents

Contents

Computer Applications

Program overview

This curriculum is designed as per the guidelines of the National Skills Qualification Framework (NSQF) for Level 4. The course equips learners with theory and practical skills, needed to get jobs in the field of computer science.

A learner may exit after 6 months with a certificate course, or after 1 year with a diploma course or may continue further for advanced diploma or B.Voc. degree.

Job Opportunities:

After completing the 6-month certificate course in computer applications, students will be equipped with the technical foundation needed to pursue a variety of entry-level job opportunities in both IT and administrative roles. The course focuses on key areas such as Python programming, software testing, office tools, and computer architecture, making students well-prepared for specialized roles within the IT sector.

In the field of IT Support and Technical Assistance, students will be ready for roles such as Help Desk Executive or Technical Support Assistant. With troubleshooting skills and a solid understanding of computer architecture, they will assist users with resolving software issues and managing technical problems. Students can also pursue positions as Computer Lab Assistants, where their combined knowledge of both software and hardware will help them support users in educational or corporate environments by troubleshooting and assisting with basic computing tasks.

The course will also prepare students for various Office and Administrative Roles. Proficiency in tools like MS Office (Word, Excel, PowerPoint) will make students ideal candidates for positions such as Data Entry Operators or Office Assistants, where they will be responsible for managing documents, organizing data, and creating reports. Furthermore, students can pursue opportunities as E-commerce Executives, where they will use their skills to manage product listings, track stock, and assist in the day-to-day operations of online sales platforms.

Freelance and remote work opportunities will also be available for students. With their proficiency in office tools and data management, students can easily transition into Freelance Data Entry or Content Editing roles. Additionally, the basic programming knowledge gained in Python during the course will open doors for Freelance Software Development work, including web development, application development, and basic scripting tasks.

Students can also explore roles in Social Media and Digital Marketing. With technical skills to manage campaigns, analyze data, and optimize online platforms, they will be well-positioned to work as Digital Marketing Assistants or Social Media Managers.

One of the primary areas for employment will be in Quality Assurance (QA) and Software Testing. With the knowledge gained from testing methodologies and basic programming skills, students can step into roles such as Test Engineer for product development, where they will perform manual and automated testing, report bugs, and contribute to the overall quality of software products. They can also pursue positions as Test Automation Engineers, where they will utilize Python and automation tools to write test scripts and build frameworks for automating repetitive tasks. Additionally, students will be prepared for Manual Tester roles, ensuring that software applications meet required standards through test case execution and bug documentation.

By completing this course, students will gain the technical expertise required for a variety of roles in the IT-ITeS sector, such as software testing, IT support, and basic software development. They will also be ready for administrative, e-commerce, and freelance opportunities, ensuring a broad range of career paths in the ever-evolving tech industry and learners receive career readiness support. The hands-on experience with programming, testing, and office tools ensures they are prepared for both entry-level technical and support positions.

Proposed Student Intake: 30 (may be increased later on).

Minimum Educational Qualification and Experience: 12th grade Pass - with computer background,10th grade pass and 2 Years Experience - in relevant field, Previous relevant Qualification of NSQF Level - 3- and 3-Years Experience - in relevant field.

Duration: 6 months.

Career Path:
With a Six months certificate course can work in the IT sector or as a freelancer or IT support staff in any sector. The willing candidate may opt for higher education in Computer Applications, i.e. Diploma in Computer Application (Level: 05), Advanced Diploma in Computer Application (Level: 06), B. Voc. Degree in Computer Application (Level: 07), Post Graduate Diploma in Computer Application (Level: 08), M. Voc. Degree in Computer Application (Level: 09), and Research in Computer Application (Level:10).

Curriculum/Syllabus

This program is aimed at training candidates for the job of a "Computer Application", in the "IT-ITeS" Sector/Industry and aims at building the following key competencies amongst the learner.

Training Delivery Plan	
Program Name	**Computer Application**
Qualification Pack and Ref Id	**SSC/Q7001**
Prerequisite to the Training	12th grade Pass - with computer background,10th grade pass and 2 Years Experience - in relevant field, Previous relevant Qualification of NSQF Level - 3- and 3-Years Experience - in relevant field.
Training Outcome	**By the end of this program, the participants would have achieved the following competencies:** **1. Apply Proficiency in Basic Office Tools** • Create and edit professional documents, spreadsheets, and presentations using MS Office or Google Workspace. • Conduct web searches efficiently and evaluate the credibility of digital sources. • Design forms, quizzes, and social media content for data collection and digital outreach. **2. Demonstrate Foundational Programming Skills Using Python** • Write, debug, and run Python programs involving variables, data types, control structures, and functions. • Manipulate data using built-in data structures (lists, dictionaries, tuples, sets). • Handle file operations and exceptions in Python, with practical application in basic problem-solving tasks. **3. Implement Core Programming Concepts** • Solve real-world problems using Python logic, algorithms, and modular coding practices. • Apply loops, conditionals, error handling, and debugging techniques for developing clean and

	efficient code. **4. Understand Computer Architecture and Digital Logic** • Explain the structure and function of key hardware components including CPU, memory, ALU, and I/O units. • Simulate CPU instructions and understand memory reference operations and instruction cycles. • Design and interpret combinational and sequential digital circuits. **5. Apply Software Engineering and Quality Assurance Principles** • Understand and apply various software development life cycles (Waterfall, Agile). • Write and execute test cases to ensure software functionality, using both manual and automated testing methods. • Track, report, and resolve software defects while emphasizing software quality and continuous integration (CI) practices. **6. Perform Manual and Automated Software Testing** • Design, implement, and automate test scripts using tools such as **Pytest**. • Detect, log, and communicate bugs with clarity and completeness. • Integrate testing workflows into development cycles for early error detection and improved quality. **7. Communicate Effectively in Technical Environments** • Write clear, grammatically accurate technical documents including reports, manuals, and proposals. • Summarize technical content, write professional emails, and follow business communication etiquette. • Collaborate on writing tasks, presentations, and digital content development for technical audiences. **8. Prepare for Diverse Job Roles in IT and Related Fields**

	• Be job-ready for roles such as:
	  ○ Python Programmer (Entry-Level)
	  ○ Manual/Automation Test Engineer
	  ○ Help Desk Executive / Technical Support Assistant
	  ○ Data Entry Operator / Office Assistant
	  ○ Digital Content Assistant / Social Media Executive
	  ○ Freelance Software Developer / E-commerce Executive

Modules with NOS codes

S.No.	Module	Alignment Rationale
1	**SSC/N7013: Basic Office Tools**	Prepares learners to create, edit and share professional documents, spreadsheets and presentations; aligns with documenting and reporting test artefacts.
2	**SSC/N7015: Introduction to Python Language**	Python skills are used to create scripts/utilities for test case execution and automation.
3	**SSC/N7015: Programming Fundamentals**	Strengthens programming logic and data handling to develop automated tests and tools.
4	**SSC/N7015: Computer Architecture and Networking & Databases**	Understanding of hardware & instruction cycles supports effective test execution and defect analysis. Understanding of **computer networking basics**: LAN, WAN, IP addressing, TCP/IP protocols.
5	**SSC/N7011: Software Engineering & Quality Assurance**	Covers SDLC, QA concepts, continuous integration, and managing defects.
6	**SSC/N7012: Introduction to Software Testing**	Directly matches core testing skills—manual & automated test execution, defect logging.
7	**SSC/N7016: Technical Communication**	Focuses on clear reporting, technical documentation, and professional communication of results.
8	**SSC/N7016: Cybersecurity Fundamentals**	Cybersecurity aligns with workplace safety and secure coding standards.
9	**SSC/N7014: Career Services & Placement Preparation**	Career Services aligns with communication, coordination, and professionalism NOS.

Note: “Certain NOS codes are supported across multiple modules to ensure adequate depth and practice (e.g., SSC/N7015 is covered in Modules 2, 3 and 4).”

Details of Modules

S.No.	Module	Key Learning Outcomes	Equipment Requirements
1	**Basic Office Tools** **Theory Duration (hh:mm):** 30:00 **Practical Duration (hh:mm)**: 45:00	• Learn to effectively use Search Engines: Participants will develop the ability to use search techniques and keywords to find accurate, relevant, and trustworthy information online. • Clear, concise, and effective emails to enhance communication: They will practice writing well-structured emails that are professional, to the point, and suitable for academic or workplace settings. • Develop the skills to create, format, and edit professional documents efficiently using word processing tools: Learners will gain hands-on experience in drafting, editing, and formatting documents using tools like MS Word or Google Docs. • Learn to design impactful, visually engaging presentations that effectively communicate your message: They will be able to use presentation tools like PowerPoint or Google Slides to design clear, well-structured, and appealing slides. • Gain proficiency in organizing, analyzing, and visualizing data using spreadsheet software: Participants will work with Excel or Google Sheets to manage data, apply formulas, and create charts	• Laptop • Internet

		for meaningful insights. ● Master the use of Google Forms to create surveys, quizzes, and collect data seamlessly: They will learn to design user-friendly forms for feedback, assessments, or data collection with automated responses. ● Acquire the knowledge to create and share engaging multimedia content across various social media platforms: Learners will explore tools and techniques to produce content such as videos, posters, and posts that are suitable for online sharing.	
2	**Introduction to Python Language** **Theory Duration (hh:mm):** 30:00 **Practical Duration (hh:mm)**: 45:00	● Understand the basics of Python programming, including the Python interpreter, and the IDLE user interface: Participants will be introduced to the Python environment, learning how to write and run code using the Python interpreter and IDLE. ● Learn to use Python as a calculator and perform simple operations with numeric data types such as literals, expressions, and comparisons: They will perform arithmetic operations and logical comparisons using Python's built-in numeric data types. ● Gain a foundational understanding of different data types like numbers, strings, lists, dictionaries, and tuples: Learners will explore Python's core data types and	● Laptop ● Internet ● Python Programming Language ● Cloud IDEs such as **Replit** or **VS Code Online** to allow coding without local installation.

		understand how to store and organize different kinds of information. ● Learn to manipulate and format strings, perform various list operations, and work with dictionaries for data storage: They will develop skills to process text, modify sequences, and store key-value pairs using built-in functions and methods. ● Master file handling operations, including creating, opening, and modifying files: Participants will learn to read from and write to files, enabling them to store and retrieve data effectively in Python programs. ● Use cloud-based IDEs (Replit/VSCode Online) to write, run, and share Python programs.	
3	**Programming Fundamentals** **Theory Duration (hh:mm):** 30:00 **Practical Duration (hh:mm)**: 45:00	● Learn how to use core Python programming concepts in practical programming scenarios: Participants will apply basic Python concepts to build simple programs that solve everyday problems. ● Learn to implement logical decision-making in Python programs using control structures such as conditionals (if, else) and loops (for, while), ensuring efficient program execution: They will gain the ability to control the program flow using conditions and repetitions, making the code	● Laptop ● Internet ● Python Programming Language ● Git, GitHub account setup, and internet access.

		dynamic and responsive. ● Gain proficiency in Python's built-in data structures like lists, dictionaries, sets, and tuples to store, manage, and manipulate data effectively: Learners will understand how to choose and use the right data structure to efficiently organize and process information. ● Understand how to define and use functions in Python to modularize code, improving reusability, maintainability, and readability: They will learn to write reusable blocks of code using functions, making programs easier to debug and manage. ● Learn essential techniques for error handling in Python using try-except blocks and develop skills for debugging and troubleshooting to write clean, efficient code: Participants will be equipped to anticipate and manage errors in their code and use debugging techniques to fix issues effectively. ● Develop the ability to break down complex problems and implement algorithms in Python, using built-in libraries and writing custom solutions to solve real-world challenges: They will practice solving practical problems by designing step-by-step logic and implementing it through efficient Python code. ● Apply version control using Git and GitHub for managing, sharing, and collaborating on Python	

		projects.	
4	**Computer Architecture and Networking & Databases** **Theory Duration (hh:mm):** 40:00 **Practical Duration (hh:mm)**: 55:00	• Understand the basic components of digital computers: Participants will learn about key digital components such as input/output units, ALU, memory, and control units that form the foundation of a computer system. • Learn to design combinational and sequential circuits: They will explore how to build logical circuits that perform specific functions, both without memory (combinational) and with memory (sequential). • Gain proficiency in the use of integrated circuits and digital components: Learners will understand how integrated circuits function and how to use them effectively in the construction of digital systems. • Learn about memory units and their applications in digital systems: Participants will study various types of memory (RAM, ROM, cache) and their roles in data storage and processing within computer systems. • Learn the architecture of computers, including instruction codes, computer instructions, and the timing and control of the instruction cycle: They will understand how computers execute instructions step-by-step and how instruction timing and control	• Laptop • Internet • CPU Sim software • Laptop with internet access. • Networking tools (ping, traceroute, nslookup, Wireshark or Packet Tracer). • Database setup (MySQL/PostgreSQL/SQLite). • Client-server simulation software or local server environment.

		signals work during processing. ● Understand memory reference instructions and their implementation using arithmetic, logical, program control, and input/output micro-operations: Learners will see how basic computer instructions manipulate data in memory using operations like addition, comparison, jumps, and I/O. ● Gain knowledge of various instruction formats, including zero-address, one-address, two-address, and three-address instructions. Learn how different instruction formats impact the overall design and efficiency of a CPU: They will explore how instruction formats determine operand access and how this choice influences the performance and structure of the processor. ● Simulating basic instructions on a CPU simulator is the ability to execute and analyze fundamental CPU instructions, understanding their impact on registers, memory, and program flow: Participants will use simulation tools to observe how instructions affect a computer's internal state, helping them visualize program execution at the hardware level. ● Understand computer networking basics: LAN, WAN, IP addressing, TCP/IP protocols. ● Explain client-server architecture and its role in applications.	

		• Perform basic network troubleshooting (ping, traceroute, DNS, firewall, connectivity issues). • Describe database concepts: tables, rows, columns, keys, relationships. • Write and execute basic SQL queries (SELECT, INSERT, UPDATE, DELETE). • Demonstrate database backup, restore, and security practices.	
5	**Software Engineering and Quality Assurance** **Theory Duration (hh:mm):** 15:00 **Practical Duration (hh:mm)**: 25:00	• Gaining proficiency in writing and debugging code using programming languages such as Python: Participants will learn to write efficient code and troubleshoot errors to build functional software applications. • Understanding and applying various software development life cycle (SDLC) models, including Agile and Waterfall: They will explore structured approaches to software development and learn how different models guide project planning and execution. • Developing skills in designing, executing, and automating software tests to ensure product quality: Learners will understand how to verify software functionality through manual and automated testing techniques. • Gaining hands-on experience in identifying, reporting, and resolving software defects: They will practice detecting bugs,	• Laptop • Internet • Python Programming Language • Git, GitHub, Selenium framework.

		documenting them clearly, and working toward effective resolutions. ● Mastering test case creation, test execution, and defect management using QA tools and methodologies: Participants will become familiar with writing test cases, running tests, and using tools to track and manage software defects. ● Understanding the importance of continuous integration and delivering high-quality software through effective collaboration and feedback loops: They will learn how regular code integration, team coordination, and feedback improve software quality and project success. ● Use Git/GitHub for version control, collaboration, and maintaining project repositories." ● Apply automation tools such as **Selenium** for functional and regression testing.	
6	**Introduction to Software Testing** **Theory Duration (hh:mm):** 15:00 **Practical Duration (hh:mm)**: 25:00	● Understanding the fundamentals of software testing and its importance in the development process: Participants will learn the role of testing in ensuring software reliability, performance, and user satisfaction. ● Learning how to design and write test cases to validate software functionality: They will gain the ability to create detailed test scenarios that check whether the	● Laptop ● Internet ● Python Programming Language ● Pytest ● Selenium WebDriver, compatible browsers (Chrome/Firefox).

		software behaves as expected. ● Gaining hands-on experience in using Pytest for automating test execution and reporting results: Learners will use Pytest to automate testing tasks, execute multiple tests efficiently, and generate test reports. ● Developing skills in identifying and documenting software defects to improve product quality: They will practice detecting errors in software behavior and clearly recording them to aid in fixing issues. ● Understanding the integration of automated tests into the software development lifecycle for continuous testing and improvement: Participants will see how automated testing fits into development workflows, helping teams catch bugs early and maintain code quality. ● Automate browser-based test cases using Selenium.	
7	**Technical Communication** **Theory Duration (hh:mm):** 15:00 **Practical Duration (hh:mm)**: 55:00	● Developing effective reading strategies, such as skimming and scanning, to enhance comprehension of technical texts and manuals: Participants will learn to quickly locate key information and grasp the overall meaning of complex technical documents. ● Mastering the ability to summarize and paraphrase technical content	● Laptop ● Internet ● **Git & GitHub account** (for version control and portfolio hosting). ● **Cloud IDEs** (Replit, VS Code Online) or local IDEs

		accurately while identifying main ideas and supporting details: They will practice condensing information and rephrasing content clearly without losing its original meaning. • Gaining proficiency in technical writing by producing clear, concise instructions, and well-structured technical reports and proposals: Learners will develop the skills to write precise technical documents that convey information logically and effectively. • Improving business communication skills by writing professional emails, letters, and adhering to proper etiquette and tone in correspondence: They will enhance their ability to communicate clearly and professionally in workplace settings through written communication. • Enhancing grammar and sentence structure knowledge to avoid common errors, and effectively use tenses, voice, and modals in technical writing: Participants will strengthen their grammar usage to create accurate, polished, and technically sound documents. • Acquiring advanced technical writing skills, including writing user manuals, creating digital content for websites, and mastering collaborative writing, critical analysis, and presentation techniques for technical projects: They will explore advanced forms of technical documentation and	(PyCharm, VS Code). • **Python environment** (installed locally or via cloud IDE). • **Office productivity tools** (MS Office / Google Workspace). • **Test automation tools**: Pytest and Selenium (for uploading test scripts). • **Browser with GitHub access** (Chrome/Firefox)

		develop skills for teamwork, content creation, and presenting complex information clearly. • Maintain a personal GitHub repository to document and showcase learning progress. • Upload and organize Python scripts (from Modules 2 & 3). • Publish test cases and automated scripts (from Modules 5 & 6). • Share Office Tools outputs such as formatted documents, spreadsheets with analysis, and presentation decks (from Module 1). • Document project work and reports with proper README files and version control practices. • Demonstrate ability to present the GitHub portfolio during evaluation.	
8	**Cybersecurity Fundamentals** **Theory Duration (hh:mm):** 05:00 **Practical Duration (hh:mm)**: 10:00	• Understand principles of **safe coding practices** to avoid common vulnerabilities (e.g., input validation, error handling, avoiding hard-coded credentials). • Identify and prevent **phishing attacks** through email hygiene, link verification, and reporting suspicious activity. • Apply **data protection practices**, including secure passwords, encryption basics, and responsible handling of personal and organizational data. • Demonstrate the ability to configure basic security settings on devices and applications.	• Laptop with internet access • Email client/web browser (for phishing simulation) • Open-source cybersecurity awareness tools (e.g., HaveIBeenPwned, phishing simulators)

9	**Career Services & Placement Preparation** **Theory Duration (hh:mm):** 05:00 **Practical Duration (hh:mm)**: 15:00	• Prepare a professional resume and LinkedIn profile tailored for IT and QA roles. • Practice mock technical interviews focusing on Python, QA, and IT support scenarios. • Attempt mock coding tests and aptitude assessments similar to entry-level recruitment exams. • Gain exposure to group discussions and communication rounds for client-facing roles. • Build confidence in presenting GitHub portfolio projects during interviews.	• Laptop with internet access. • Online coding platforms (HackerRank , CodeSignal, or equivalent). • Resume templates and LinkedIn access. • Mock interview tools (Google Meet/Zoom for practice sessions).
	Total Duration (hh:mm): 450:00 **Theory Duration (hh:mm):** 180:00 **Practical Duration (hh:mm)**: 270:00		

Credits	Total Notional Hours	Hours per Credit
15	**450**	**1 Credit = 30 hrs (NCrF). ABC Compliant**

Module wise PC, KU, GE and Outcomes

◇ Module 1: Basic Office Tools

Knowledge and Understanding (KU)

- KU1: Principles of digital literacy and information search techniques.
- KU2: Structure, features, and functions of word processing, spreadsheet, and presentation tools (MS Office / Google Workspace).
- KU3: Basics of email communication, formatting, tone, and workplace etiquette.
- KU4: Data organization, use of formulas, and visualization techniques in spreadsheets.
- KU5: Fundamentals of online survey and form design (Google Forms or equivalent).
- KU6: Basics of multimedia content creation for digital and social platforms.
- KU7: Best practices for accuracy, clarity, and consistency in documentation.

Generic Skills (GS)

- GS1: Communicate effectively in written English (emails, reports, presentations).
- GS2: Apply critical thinking to evaluate the credibility of online sources.
- GS3: Organize and present information clearly using digital tools.
- GS4: Demonstrate creativity in designing presentations, forms, and social media content.
- GS5: Collaborate with peers through shared digital documents and online platforms.
- GS6: Practice time management by preparing documents and presentations within deadlines.
- GS7: Use problem-solving skills to troubleshoot formatting, data errors, or layout issues in office applications.

Learning Outcomes of the Module

By the end of this module, the learner will be able to:

1. Use search engines effectively and evaluate the reliability of online information.
2. Draft professional emails and written communication suitable for workplace settings.
3. Create, format, and edit structured documents using word processing tools.
4. Design visually appealing and clear presentations aligned with audience needs.
5. Use spreadsheets to organize, analyze, and visualize data for meaningful insights.
6. Develop online forms/quizzes to collect and manage data efficiently.
7. Create and share engaging multimedia and social media content.

Note: The outcomes are the same as mentioned above. But they are restructured into simple format here.

Module 2: Introduction to Python Language

Knowledge & Understanding (KU)

- KU1: Python interpreter, IDE, and environment basics.
- KU2: Data types: numbers, strings, lists, tuples, dictionaries.
- KU3: File operations: read/write/modify.
- KU4: Arithmetic and logical operators.
- KU5: Built-in functions for string and list manipulation.

Generic Skills (GS)

- GS1: Apply logical reasoning in coding tasks.
- GS2: Practice accuracy and debugging discipline.
- GS3: Communicate code purpose and logic effectively.
- GS4: Use self-learning to explore Python libraries.

Learning Outcomes

1. Run and test Python programs using an IDE.
2. Perform arithmetic/logical operations with Python data types.
3. Manipulate strings, lists, and dictionaries.
4. Apply file handling in coding tasks.
5. Build simple Python scripts for real-world applications.

Note: The outcomes are the same as mentioned above. But they are restructured into simple format here.

Module 3: Programming Fundamentals

Knowledge & Understanding (KU)

- KU1: Control structures (if, else, loops).
- KU2: Modular programming using functions.
- KU3: Data structures: lists, sets, tuples, dictionaries.
- KU4: Error handling using try-except.
- KU5: Debugging tools and techniques.
- KU6: Algorithm design and problem-solving methods.

Generic Skills (GS)

- GS1: Analyze problems logically and break them into smaller parts.
- GS2: Apply systematic debugging and error handling.
- GS3: Use programming discipline for modular, reusable code.
- GS4: Communicate problem-solving approaches effectively.

Learning Outcomes

1. Write Python programs using loops and conditionals.
2. Organize and manipulate data with built-in structures.
3. Define and use modular functions.
4. Apply error handling and debugging practices.
5. Solve practical problems through algorithmic coding.

Note: **The outcomes are the same as mentioned above. But they are restructured into simple format here.**

◇ Module 4: Computer Architecture and Networking & Databases

Knowledge & Understanding (KU)

- KU1: Components of digital computers: CPU, ALU, memory, I/O.
- KU2: Basics of combinational and sequential circuits.
- KU3: Boolean algebra and simplification techniques.
- KU4: Memory types and organization.
- KU5: Instruction cycle, codes, and formats.
- KU6: Micro-operations and CPU execution process.
- KU7: Principles of networking and common network protocols.
- KU8: Client-server communication in modern applications.
- KU9: Relational database structure and terminology.
- KU10: SQL basics for querying and manipulating data.
- KU11: Fundamentals of data security in databases.

Generic Skills (GS)

- GS1: Apply analytical skills to interpret digital circuit designs.
- GS2: Use logical reasoning to connect hardware concepts with programming.
- GS3: Communicate architecture diagrams and processes clearly.
- GS4: Apply logical reasoning to troubleshoot network issues.
- GS5: Demonstrate systematic problem-solving in SQL tasks.
- GS6: Communicate technical solutions clearly to peers/users.
- GS7: Collaborate on small IT support projects using networking and database knowledge.

Learning Outcomes

1. Explain CPU, memory, and I/O components.
2. Design and interpret simple digital circuits.
3. Analyse Boolean expressions and optimize circuits.
4. Demonstrate instruction execution in a CPU simulator.
5. Connect architecture concepts to software program execution.
6. Understand computer networking basics: LAN, WAN, IP addressing, TCP/IP protocols.
7. Explain client-server architecture and its role in applications.
8. Perform basic network troubleshooting (ping, traceroute, DNS, firewall, connectivity issues).
9. Describe database concepts: tables, rows, columns, keys, relationships.
10. Write and execute basic SQL queries (SELECT, INSERT, UPDATE, DELETE).
11. Demonstrate database backup, restore, and security practices.

Note: **The outcomes are the same as mentioned above. But they are restructured into simple format here.**

◇ Module 5: Software Engineering & Quality Assurance

Knowledge & Understanding (KU)

- KU1: SDLC models – Waterfall, Agile.
- KU2: Testing principles and QA standards.
- KU3: Manual and automated test design concepts.
- KU4: Bug reporting and defect tracking methods.
- KU5: Continuous Integration (CI) practices.

Generic Skills (GS)

- GS1: Collaborate effectively in QA teams.
- GS2: Write clear defect reports and documentation.
- GS3: Apply critical thinking to assess software quality.
- GS4: Manage tasks systematically within a development cycle.

Learning Outcomes

1. Explain SDLC models and QA processes.
2. Design and execute test cases.
3. Track and report defects accurately.
4. Apply CI practices to improve quality.
5. Integrate QA processes into software projects.

Note: **The outcomes are the same as mentioned above. But they are restructured into simple format here.**

◇ Module 6: Introduction to Software Testing

Knowledge & Understanding (KU)

- KU1: Fundamentals of software testing and its role.
- KU2: Test case writing methodology.
- KU3: Test automation concepts (Pytest).
- KU4: Defect identification and documentation.
- KU5: Integration of testing into SDLC.

Generic Skills (GS)

- GS1: Apply systematic thinking in test design.
- GS2: Communicate findings clearly in bug reports.
- GS3: Collaborate with developers to improve code quality.
- GS4: Apply discipline in executing test workflows.

Learning Outcomes

1. Describe purpose and scope of testing.
2. Create test cases for functionality validation.
3. Use Pytest for automated test execution.
4. Detect and document defects.
5. Integrate testing practices into development workflows.

Note: The outcomes are the same as mentioned above. But they are restructured into simple format here.

◇ Module 7: Technical Communication

Knowledge & Understanding (KU)

- KU1: Reading and summarizing technical texts.
- KU2: Grammar, sentence structure, and business writing standards.
- KU3: Technical writing formats: manuals, reports, proposals.
- KU4: Professional email and workplace etiquette.
- KU5: Collaborative documentation practices.
- KU6: Basics of Git and GitHub workflows (init, commit, push, pull, branch).
- KU7: Importance of maintaining a digital portfolio for career progression.
- KU8: Professional standards for documenting and sharing code/projects.

Generic Skills (GS)

- GS1: Communicate clearly in written English.
- GS2: Summarize and paraphrase technical information.
- GS3: Collaborate effectively in team writing tasks.
- GS4: Apply critical thinking to analyze and present information.
- GS5: Apply self-management to update the portfolio regularly.
- GS6: Demonstrate digital professionalism through clean code and clear documentation.
- GS7: Communicate technical work effectively using README files and structured repositories.
- GS8: Collaborate via GitHub by sharing repositories and reviewing contributions.

Learning Outcomes

1. Apply reading strategies to understand technical material.
2. Write structured reports, manuals, and proposals.
3. Compose professional business emails and letters.
4. Summarize and paraphrase content accurately.
5. Collaborate on technical documents and presentations.
6. Maintain a personal GitHub repository to document and showcase learning progress.
7. Upload and organize Python scripts (from Modules 2 & 3).
8. Publish test cases and automated scripts (from Modules 5 & 6).
9. Share Office Tools outputs such as formatted documents, spreadsheets with analysis, and presentation decks (from Module 1).
10. Document project work and reports with proper README files and version control practices.
11. Demonstrate ability to present the GitHub portfolio during evaluation.

Note: The outcomes are the same as mentioned above. But they are restructured into simple format here.

Module 8: Cybersecurity Fundamentals

Key Learning Outcomes

- Understand principles of **safe coding practices** to avoid common vulnerabilities (e.g., input validation, error handling, avoiding hard-coded credentials).
- Identify and prevent **phishing attacks** through email hygiene, link verification, and reporting suspicious activity.
- Apply **data protection practices**, including secure passwords, encryption basics, and responsible handling of personal and organizational data.
- Demonstrate the ability to configure basic security settings on devices and applications.

Knowledge and Understanding (KU)

- KU1: Common cybersecurity threats (phishing, malware, social engineering).
- KU2: Safe coding guidelines (validation, sanitization, secure storage).
- KU3: Data privacy concepts (confidentiality, integrity, availability).
- KU4: Basics of data encryption and password management.

Generic Skills (GS)

- GS1: Practice security hygiene in professional and personal digital use.
- GS2: Apply critical thinking to recognize suspicious communication.
- GS3: Communicate effectively about security risks to team members.
- GS4: Demonstrate responsible behavior in handling sensitive data.

Module 9: Career Services & Placement Preparation

Key Learning Outcomes

- Prepare a professional resume and LinkedIn profile tailored for IT and QA roles.
- Practice mock technical interviews focusing on Python, QA, and IT support scenarios.
- Attempt mock coding tests and aptitude assessments similar to entry-level recruitment exams.
- Gain exposure to group discussions and communication rounds for client-facing roles.
- Build confidence in presenting GitHub portfolio projects during interviews.

Knowledge and Understanding (KU)

- KU1: Standard resume formats and industry expectations for IT job seekers.
- KU2: Common coding test patterns (Python, logic, SQL basics).
- KU3: Technical interview structure and frequently asked QA/programming questions.
- KU4: Importance of soft skills, confidence, and clarity in placement processes.

Generic Skills (GS)

- GS1: Communicate clearly in professional interviews and group discussions.
- GS2: Apply problem-solving under time-bound coding test conditions.
- GS3: Demonstrate self-management and confidence during assessments.
- GS4: Present personal projects effectively as part of a portfolio.

Trainer Prerequisites for Job role: "Computer Application" mapped to Qualification Pack: "SSC/Q7001, v1.0"

S. No.	Area	Details
1	**Description**	The trainer should have in-depth knowledge of software testing, Python programming, office productivity tools, and quality assurance processes relevant to the IT-ITeS sector.
2	**Personal Attributes**	Good communication and interpersonal skills, ability to engage learners, patience and empathy, strong analytical/problem-solving skills, and a commitment to continuous learning.
3	**Minimum Educational Qualifications**	Graduate in Computer Science/IT (BCA, B.Sc. IT, B.Tech) or equivalent. Higher qualification (MCA/M.Tech) preferred.
4	**Domain Certification**	Certified in relevant areas such as Software Testing (ISTQB Foundation/Advanced), Python Programming, or Quality Assurance certifications recognized by SSC NASSCOM or industry bodies and certified for SSC/Q7001 job role.
5	**Platform Certification**	Mandatory ToT (Training of Trainers) certification under SSC NASSCOM/NSDC guidelines. Experience with Learning Management Systems (LMS) preferred.
6	**Experience**	Minimum 3 years of industry experience in software testing, QA, or application development. At least 1 year of experience in training/teaching in IT-ITeS domain.

Trainer CPD Requirement

- Trainers delivering the *Computer Application* curriculum must complete **a minimum of 16 hours of CPD annually**.
- CPD should focus on **emerging technologies and updated industry practices**, specifically:
    - **AI-based testing tools** (for test automation, bug prediction, intelligent QA).
    - **Python 3.12 updates and new features** relevant to programming and scripting.
    - **Agile QA practices** (continuous testing, DevOps integration, CI/CD workflows).
- Evidence of CPD (certificates, workshop participation, or self-learning logs) must be maintained and may be verified by SSC/affiliating body.
- Refresher hours can be accumulated through **online workshops, industry seminars, webinars, or SSC-recognized ToT update programs**.

Assessor Prerequisites for Job role: "Computer Application" mapped to Qualification Pack: "SSC/Q7001, v1.0"

1.	**Experience as Assessor**	NSQF certified assessor with ≥5 years IT-QA experience.

Assessment Criteria

S. No.	Module	Weightage
1	Basic Office Tools	15%
2	Introduction to Python Language	20%
3	Programming Fundamentals	20%
4	Computer Architecture and Networking & Databases	10%
5	Software Engineering and Quality Assurance	10%
6	Introduction to Software Testing	10%
7	Technical Communication	5%
8	Cybersecurity Fundamentals	5%
9	Career Services & Placement Preparation	5%
Total		100%

Passing Criteria: Minimum of 50% in each module and an overall score of 70% for certification.

Guidelines for Assessment:
1. Criteria for assessment for each Qualification Pack will be created by the Sector Skill Council. Each Performance Criteria (PC) will be assigned marks proportional to its importance in NOS. SSC will also lay down proportion of marks for Theory and Skills Practical for each PC.
2. The assessment for the theory & Practical part will be based on knowledge bank of questions created by the AA and approved by SSC.
3. Individual assessment agencies will create unique question papers for theory part for each candidate at each examination/training center (as per assessment criteria below)
4. Individual assessment agencies will create unique evaluations for skill practical for every student at each examination/training center based on these criteria.
5. To pass the Qualification Pack, every trainee should score a minimum of 70% cumulatively (Theory and Practical) 50% in each module.

Assessment Criteria (Module Wise):

Module 1	**Assessment Criteria**	**Marks Allocation**

		Out of	Theory	Practical
Basic Office Tools	PC1: Demonstrate the ability to use advanced search techniques to find relevant and accurate information.	10	30	120
	PC2: Evaluate the credibility and reliability of online sources.	10		
	PC3: Write professional, clear, and concise emails with appropriate tone and structure.	15		
	PC4: Create and format professional documents (e.g., reports, proposals) using word processing software.	15		
	PC5: Edit documents for clarity, grammar, and conciseness, following proper formatting standards.	10		
	PC6: Design visually appealing presentations with clear structure and minimal text.	15		
	PC7: Enhance message delivery and ensure presentations are aligned with the audience's needs, keeping them engaging and informative.	15		
	PC8: Organize data effectively in spreadsheet software, using appropriate tools like tables, charts, and graphs.	15		
	PC9: Analyze data to identify trends, patterns, or insights relevant to the task.	15		
	PC10: Design user-friendly surveys or quizzes using Google Forms, including appropriate question types	15		
	PC11: Create engaging multimedia content (videos, images, infographics) tailored for social media platforms.	15		

Total	150	30	120

Module 2 **Introduction to Python Language**	**Assessment Criteria**	**Marks Allocation**		
		Out of	**Theory**	**Practical**
	PC1: Identify and explain the basic structure of a Python program	15	60	90
	PC2: Work with numeric data types (integers, floats) to handle mathematical expressions.	20		
	PC3: Write Python expressions involving variables and constants to evaluate mathematical problems.	20		
	PC4: Identify and demonstrate the use of different data types in Python (numbers, strings, lists, dictionaries, and tuples).	20		
	PC5: Write Python code to manipulate strings (concatenate, slice, format) and apply string methods.	20		
	PC6: Perform list operations (adding/removing elements, indexing, slicing, iteration) in Python.	20		
	PC7: Demonstrate the use of dictionaries to store key-value pairs, retrieve data, and perform basic operations like adding or updating values.	20		
	PC8: Demonstrate the ability to create, open, read, and write files in Python using built-in functions	15		
Total		150	60	90

Module 3 Programming Fundamentals	Assessment Criteria	Marks Allocation		
		Out of	Theory	Practical
	PC1: Apply core Python concepts, such as variables, loops, functions, and conditionals, to solve real-world problems.	20	60	90
	PC2: Implement Logical Decision-Making with Control Structures (Conditionals and Loops)	20		
	PC3: Demonstrate the ability to create, manipulate, and access elements of lists, dictionaries, sets, and tuples.	20		
	PC4: Define Python functions with parameters and return values to modularize code for better readability and maintainability.	20		
	PC5: Demonstrate the use of built-in functions and create custom functions to solve specific programming tasks.	20		
	PC6: Implement error-handling mechanisms using try-except blocks to manage exceptions and ensure program stability.	15		
	PC7: Apply debugging techniques (e.g., print statements, Python debugger) to troubleshoot code and fix issues.	15		
	PC8: Analyze and break down complex problems into smaller, manageable tasks that can be solved using Python.	20		
Total		150	60	90

Module 4	Assessment Criteria	Marks Allocation

Computer Architecture		**Out of**	**Theory**	**Practical**
	PC1: Demonstrate knowledge of the binary system and logic gates that form the foundation of digital computing.	15	60	40
	PC2: Design and Implement Combinational and Sequential Circuits	15		
	PC3: Apply Boolean algebra and Karnaugh maps for optimizing and simplifying circuit designs.	15		
	PC4: Understand the organization and functioning of memory, including memory addressing, access times, and hierarchies.	15		
	PC5: Understand Computer Architecture, Instruction Codes, and the Instruction Cycle	10		
	PC6: Understand Memory Reference Instructions and Their Implementation Using Micro-Operations	15		
	PC7: Knowledge of Instruction Formats and Their Impact on CPU Design	15		
Total		100	60	40

Module 5 **Software Engineering and Quality Assurance**	**Assessment Criteria**	**Marks Allocation**		
		Out of	**Theory**	**Practical**
	PC1: Understanding and Applying Various Software Development Life Cycle (SDLC) Models, Including Agile and Waterfall	20	60	40
	PC2: Understanding the fundamentals of testing and types of testing	20		

	PC3: Design and write test cases to verify functionality, performance, and security requirements of software applications.	15		
	PC4: Execute test cases manually and automatically, documenting results and tracking defects.	15		
	PC5: Understanding the concept of quality assurance and its importance.	10		
	PC6: Demonstrate knowledge of continuous integration (CI) and its role in automating builds, tests, and deployments.	20		
Total		100	60	40

Module 6 **Introduction to Software Testing**	**Assessment Criteria**	**Marks Allocation**		
		Out of	**Theory**	**Practical**
	PC1: Understanding the Fundamentals of Software Testing and Its Importance in the Development Process	20	60	40
	PC2: Learning How to Design and Write Test Cases to Validate Software Functionality	20		
	PC3: Gaining Hands-On Experience in Using Pytest for Automating Test Execution and Reporting Results	30		
	PC4: Developing Skills in Identifying and Documenting Software Defects to Improve Product Quality	15		
	PC5: Understanding the Integration of Automated Tests into the Software Development Lifecycle for Continuous Testing and Improvement	15		

Total	100	60	40

Module 7 **Technical Communication**	**Assessment Criteria**	**Marks Allocation**		
		Out of	**Theory**	**Practical**
	PC1: Developing Effective Reading Strategies to Enhance Comprehension of Technical Texts and Manuals	5	30	20
	PC2: Mastering the Ability to Summarize and Paraphrase Technical Content Accurately	5		
	PC3: Gaining Proficiency in Technical Writing by Producing Clear, Concise Instructions, and Well-Structured Reports and Proposals	10		
	PC4: Improving Business Communication Skills by Writing Professional Emails, Letters, and Adhering to Proper Etiquette and Tone	10		
	PC5: Enhancing Grammar and Sentence Structure Knowledge to Avoid Common Errors in Technical Writing	10		
	PC6: Acquiring Advanced Technical Writing Skills, Including Writing User Manuals, Creating Digital Content, and Mastering Collaborative Writing Techniques	10		
Total		50	30	20

Assessment Evidence (Tangible Artefacts per Module)

Module 1: Basic Office Tools

- **Artefacts Required:**
  - Word-processed report/proposal (formatted, proofread).
  - PowerPoint/Google Slides presentation.
  - Excel/Google Sheet with formulas, charts, and analysis.
  - Google Form/Survey design.

Module 2: Introduction to Python Language

- **Artefacts Required:**
  - Python scripts demonstrating numeric operations, string/list/dictionary handling.
  - File-handling script (read/write operations).

Module 3: Programming Fundamentals

- **Artefacts Required:**
  - Python programs implementing loops, functions, and error handling.
  - Debugged code with comments.
  - Algorithm design notes/flowcharts (optional).

Module 4: Computer Architecture

- **Artefacts Required:**
  - Diagram/diagram set of CPU architecture and instruction cycle.
  - Digital logic circuit simulation (screenshots/reports from CPU Sim or equivalent).

Module 5: Software Engineering & Quality Assurance

- **Artefacts Required:**
  - Written SDLC comparison report (Waterfall vs Agile).
  - QA bug log/defect tracker (Excel or bug-tracking tool).
  - Test case document (manual/automated).

Module 6: Introduction to Software Testing

- **Artefacts Required:**
  - Pytest automated test scripts.
  - Pytest execution report (log/output).
  - Documented defect report.

Module 7: Technical Communication

- **Artefacts Required:**
  - Technical report or proposal (well-structured, formal).
  - Business email sample (professional tone).
  - User manual or digital content (short guide, website text, or collaborative writing output).