



CIRCULAR

Ref. No.: GU/Acad –PG/BoS -NEP/2025-26/148 dated 03.06.2025

In supersession to the above referred Circular, the syllabus for Semester III & IV of the **Bachelor of Vocation in Software Technologies** Programme approved by the Academic Council in its meeting held on 24th April, 12th & 13th May 2026 is attached. Further, the syllabus for Semester I & II approved earlier by the Academic Council in its meeting held on 09th May 2025 is also attached.

The Principal of the Affiliated College offering the **Bachelor of Vocation in Software Technologies** Programme is requested to take note of the above and bring the contents of the Circular to the notice of all concerned.

(Ashwin V. Lawande)
Deputy Registrar – Academic

To,

Principal of Affiliated College offering the Bachelor of Vocation in Software Technologies Programme.

Copy to:

1. Chairperson, BoS in Skill Enhancement & Vocational Studies., Goa University.
2. Controller of Examinations, Goa University.
3. Assistant Registrar Examinations (UG), Goa University.
4. Directorate of Internal Quality Assurance, Goa University for uploading the Syllabus on the University website.

Bachelor of Vocation in Software Technologies

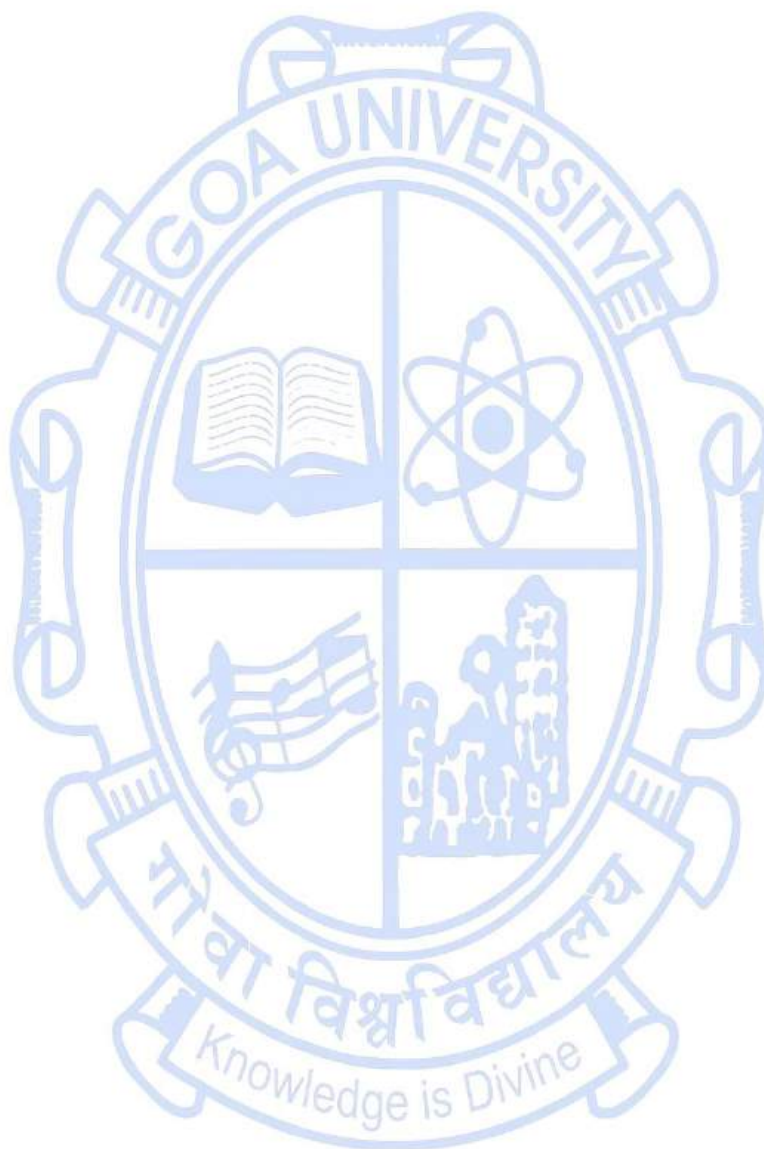
Programme Structure under NEP 2020

Certificate in Software Technologies (Semester I and Semester II)

Skill Acquired: After Successfully completing the courses of Semester-I and Semester-II, the students are expected to acquire the skills to be employable as *Software Developer*, *Desktop Publishing Associate*, and *Graphics Designer*.

NSQF /NCrF Level	Course Code	Course Title	Credits	Marks
SEMESTER-I				
Level 4.5	General Education			
	<u>STG-100</u>	Computer Fundamentals & Programming Concepts	3 (2 Theory + 1 Practical)	75
	<u>STG-101</u>	Relational Database Management System	3 (2 Theory + 1 Practical)	75
		Approved Category A - Environmental Science and Education VAC course under ordinance OA-38. (College to select any 1 course)	2 Theory	50
	Skill Development			
	<u>STG-121</u>	SSC/Q0501: Software Developer (Version 3.0)	5 Theory	300
			4 Practical	
			3 Project	
Total Semester-I		20	500	
SEMESTER-II				
Level 4.5	General Education			
	<u>STG-102</u>	Object Oriented Concepts using Java	3 (2 Theory + 1 Practical)	75
	<u>STG-103</u>	Multimedia Technologies	3 (2 Theory + 1 Practical)	75
	<u>STG-104</u>	Quantitative Techniques	2 Theory	50
	Skill Development			
	<u>STG-122</u>	SSC/Q2702: Associate – Desktop Publishing (Version 3.0)	5 Theory	300
4 Practical				

			3 Project / Internships	
	Total Semester- II		20	500
Exit Level 4.5	<u>STG-161</u>	Project Development	4 (2 Theory + 2 Practical)	100



Diploma in Software Technologies (Semester III and Semester IV)

Skill Acquired: After Successfully completing the courses of Semester-III and Semester-IV, the students are expected to acquire the skills to be employable as *User Experience Designer*, *Front-End Developer* and *Software Developer*.

NSQF / NCrF Level	Course Code	Course Title	Credits	Marks
SEMESTER-III				
Level 5	General Education			
	<u>STG-200-</u>	Web Designing Concepts	3 (2 Theory + 1 Practical)	75
	<u>STG-201</u>	Data Structures	3 (2 Theory + 1 Practical)	75
	<u>STG-202</u>	Reasoning Techniques	2 Theory	50
	Skill Development			
	<u>STG-221</u>	SSC/Q8404: User Experience Designer (Version 2.0)	5 Theory	300
			4 Practical	
			3 Project	
Total Semester-III		20	500	
SEMESTER-IV				
Level 5	General Education			
	<u>STG-203</u>	Software Testing & Quality Assurance	3 (2 Theory + 1 Practical)	75
	<u>STG-204</u>	Python Programming	3 (2 Theory + 1 Practical)	75
	<u>STG-205</u>	Statistics	2 Theory	50
	Skill Development			
	<u>STG-222</u>	SSC/Q8403: Application Developer - Web & Mobile (Version 2.0) Elective: Front-end Web Development	5 Theory	300
			4 Practical	
			3 Project / Internships	
Total Semester- IV		20	500	
Exit Level 5	<u>STG-261</u>	Web Application Development	4 (2 Theory + 2 Practical)	100

B.Voc. Degree in Software Technologies (Semester V and Semester VI)

Skill Acquired: After Successfully completing the courses of Semester-V and Semester-VI, the students are expected to acquire the skills to be employable as *Web Application Architect/Designer/Developer* and *Data Visualizer*.

NSQF / NCrF Level	Course Code	Course Title	Credits	Marks
SEMESTER-V				
Level 5.5	General Education			
	STG-300	Web App Development	3 (2 Theory + 1 Practical)	75
	STG-301	Data Analysis	3 (2 Theory + 1 Practical)	75
	STG-302	Computer Networks	2 Theory	50
	Skill Development			
	STG-321	SSC/Q8402: Application Architect - Web & Mobile (Version 2.0) Elective: Web Application Architecture	5 Theory	300
			4 Practical	
			3 Project	
	Total Semester-V		20	500
SEMESTER-VI				
Level 5.5	General Education			
	STG-303	Web Frameworks	3 (2 Theory + 1 Practical)	75
	STG-304	Design Thinking	3 (2 Theory + 1 Practical)	75
	STG-305	Cloud Computing	2 Theory	50
	Skill Development			
	STG-322	SSC/Q8104: AI - Data Scientist (Version 3.0) Elective: Visualizations	5 Theory	300
			4 Practical	
			3 Project / Internships	
	Total Semester- VI		20	500

B.Voc. Degree in Software Technologies (Semester VII and Semester VIII)

Skill Acquired: After Successfully completing the courses of Semester-VII and Semester-VIII, the students are expected to acquire the skills to be employable as **Junior Data Scientist / Data Scientist**.

NSQF / NCrF Level	Course Code	Course Title	Credits	Marks
SEMESTER-VII				
Level 6	General Education			
	STG-400	Neural Networks	3 (2 Theory + 1 Practical)	75
	STG-401	Graph Theory	3 (2 Theory + 1 Practical)	75
	STG-402	Advanced Statistics - I	2 Theory	50
	Skill Development			
	STG-421	SSC/Q8104: AI - Data Scientist (Version 3.0) Elective: Model Risk Assessment	5 Theory	300
			4 Practical	
			3 Project	
Total Semester-VII		20	500	
Level 6	General Education			
	STG-403	Agile Project Management	3 (2 Theory + 1 Practical)	75
	STG-404	Digital Marketing	3 (2 Theory + 1 Practical)	75
	STG-405	Advanced Statistics - II	2 Theory	50
	Skill Development			
	STG-422	SSC/Q8104: AI - Data Scientist (Version 3.0) Elective: Model Business Performance	5 Theory	300
			4 Practical	
			3 Project / Internships	
Total Semester- VIII		20	500	

SEMESTER-I

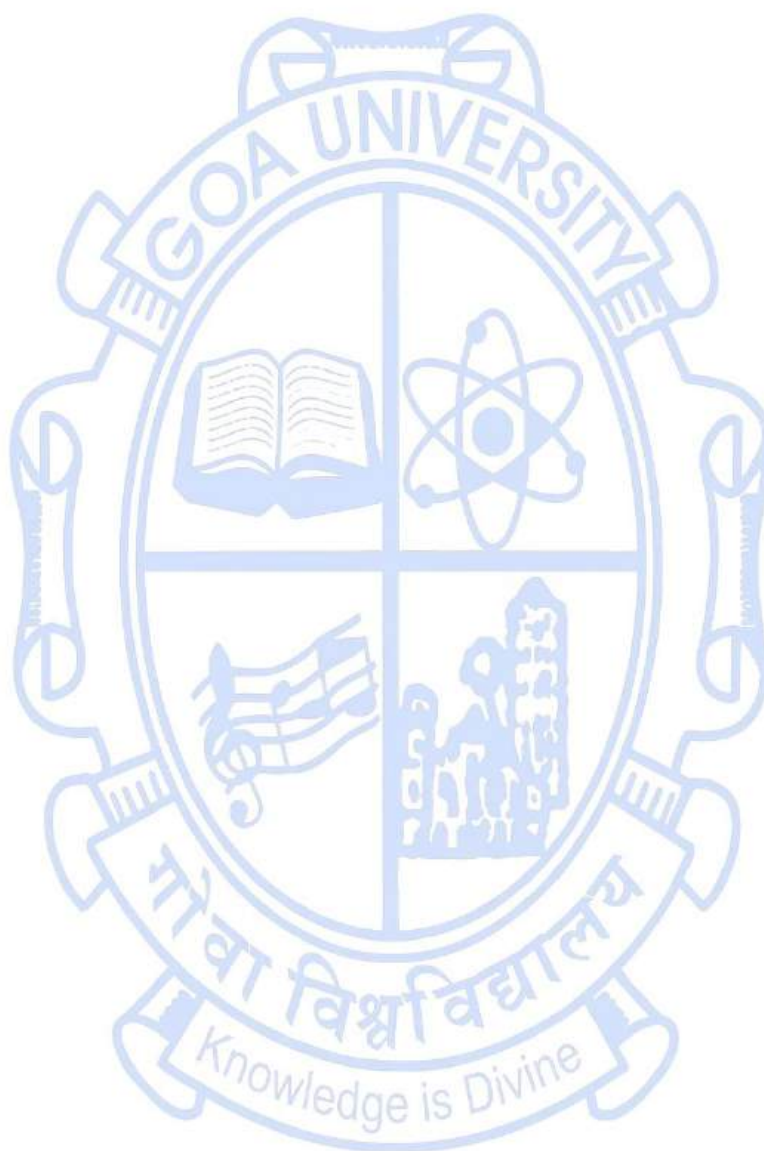
Name of the Programme	: B.Voc (Software Technologies)
Course Code	: STG-100
Title of the Course	: Computer Fundamentals and Programming Concepts
Number of Credits	: 3 (2T+1P)
Effective from AY	: 2 0 2 5 - 2 6

Pre-requisites for the Course:	None	
Course Objectives:	1. To understand the concepts and techniques of problem solving 2. To analyze, understand and build logic to solve problems 3. To design algorithms and flowcharts for better understanding and documentation for accurate implementation of the problem 4. To code and implement a well-structured, robust programming logic using C	
Content:		No of hours
Unit 1	Number System: Decimal to binary conversion and vice versa. Binary number representation (signed, 1's Complement and 2's complement), Binary Arithmetic - addition, subtraction. Introduction to Programming Concepts: Basic Programming Concepts, Steps Involved in Computer Programming – Problem Definition – Outlining the Solution – Algorithms & Flowchart, Types of Errors in programming, Translators, Interpreter & compilers, Languages Types, Programming approaches, Features of good algorithm and programming Developing algorithm, flow charts and pseudo code of different types of problems. Introduction to C Programming: Overview of C: History and importance of C, Basic structure of C program, executing a C program. Constants, Variable and Data Types: Introduction, Character Set, C Tokens, Keywords and Identifiers, Constants, Variables, Data Types, Declaration of Variables, Assigning Values to Variables, Defining Symbolic Constants. Input and Output Operations in C: Reading a Character, writing a Character, Formatted Input, Formatted Output.	15
Unit 2	Operators and Expressions: Introduction, Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Increment and Decrement Operators, Conditional Operator, Bitwise Operators, Special Operators, Arithmetic Expressions, Evaluation of Expressions, Precedence of Arithmetic Operators, Type Conversions in Expressions, Operator Precedence and Associativity. Decision Making and Branching: Introduction, Decision	15

	Making with IF Statement, Simple IF Statement, the IF-ELSE Statement, Nesting of IF-ELSE Statements, The ELSE IF Ladder, The Switch statement, The Conditional Operator, The goto statement, The while Statement, The do statement, The for statement. Arrays: Introduction To Arrays, single dimensional arrays and two-dimensional arrays, User defined Functions, function with arguments, call by value, call by reference	
Unit 3	Practical Work:	30
Week 1 & 2	Designing Algorithms and flowcharts for different problems.	04
Week 3 & 4	Programs using input/output statements and expressions	04
Week 5 to 7	Programs using decision-making constructs.	06
Week 8 to 10	Programs using loops.	06
Week 11 & 12	Programs using Arrays	04
Week 13 & 14	Programs using Functions	04
Week 15	Designing a Menu driven mini application	02
Pedagogy:	Apart from the normal teach-learning methodology, following are suggested. 1. Video /animation to explain various concepts 2. Collaborative, peer, flipped learning etc. 3. Adopt problem-based learning which foster student's analytical skills and develop design thinking skills.	
References/ Readings:	Text Books 1. Dromey, R. G. (1994) How to solve it by computers. Pearson Education. 2. Balaguruswamy, E. Programming in ANSI C (9th ed.) (2024) Tata McGraw-Hill Education. Reference Books 1. Venugopal, K. R., & Prasad, S. R. Mastering C. (2017) Tata McGraw-Hill Education. 2. Kanetkar, Y. Let us C (19th ed.). (2022) BPB Publications. 3. Venkateshmurthy, M. G. <i>Programming techniques through C</i> (1st ed.) (2002) Pearson Education. 4. Harbison, S. P., & Steele, G. L., Jr. C: (2002) A reference manual (4th ed.). Prentice Hall. 5. Klemens, B. <i>21st century C: (2014) C tips from the new school.</i> O'Reilly Media.	
Course Outcomes:	At the end of this course, the students will be able to: 1. Understand the Fundamentals of Binary Number system 2. Develop algorithms and compute the flow of the program to solve simple computational problems. 3. Design programmatic solutions to real world problems using C.	

- | | |
|--|--|
| | 4. Explain the concept of different data types, decision constructs, loops, functions and arrays in C programming. |
|--|--|

[\(Back to Index\)](#)



Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-101
Title of the Course	: Relational Database Management System
Number of Credits	: 3 (2T + 1P)
Effective from AY	: 2025-26

Prerequisites for the Course:	Nil	
Course Objectives:	1. To know the basic concepts of database management systems. 2. To understand the basic concepts of ER Diagram, Relational Schema, Normalization and Transaction Processing. 3. To apply the DBMS, ERD, Relational Schema Design, and Normalization. 4. To create database, relational tables, and manage it using SQL effectively.	
Units	Content	No of hours
I	Introduction to DBMS: Data, Database, Database system, Database Management System, File oriented systems and its limitations; Three schema, levels of Data Abstraction, Database Architecture (Internal, Conceptual, View) and Data Independence. Database Design and ER Modeling:ER Diagram Concepts & Terminologies Concept and Types of Entities, attributes, and relationship sets, Key attribute, and domain of an attribute. Degree of a relationship set, cardinalities, Total and partial participation Generalization, specialization, aggregation Structured Query Language (SQL) Basics Overview: Basic Concept of Tables, Fields, Records, and Keys. SQL Basics: SQL syntax, commands, and data types Data Definition Language (DDL): CREATE, ALTER, DROP commands Data Manipulation Language (DML): INSERT, UPDATE, DELETE, SELECT statements Basic Querying: Using SELECT to retrieve data, WHERE clause, filtering, and sorting. Basic Functions: COUNT, SUM, AVG, MIN, MAX	15

	Introduction to Relational Data Model: Relational model concepts. Characteristics of relations; Types of keys-super key, candidate key, primary key, and foreign key	
II	Relational Model Schema Design: Relational model constraints: Domain constraints, key constraints, primary and foreign key constraints, Integrity constraints, Referential integrity constraint and null values. Mapping Conceptual model into a normalized relational schema SQL Concepts Overview: Data Constraints: Primary Key, Foreign Key, Unique, Check, and Not Null constraints Joins: Inner Join, Left Join, Right Join, Full Join Grouping and Aggregation: GROUP BY, ORDER BY, HAVING clause Subqueries and Nested Queries Normalization: Anomalies in a database, Functional dependencies Normalization - 1NF, 2NF, 3NF: Rules, Needs and Process of conversion. Good and Bad decomposition, Basic Concepts of Views, Triggers, and Procedures. DBMS Users and Functions: Database Users, DBMS functions, Advantages and Disadvantages, Database Administration, Control and Functions Transaction Processing Basics: Concept and ACID Properties, state Diagram of Transactions,	15
III	List of Practical To be done using any suitable RDBMS software like MYSQL	30
Week 1 & 2	ER Diagram and Relational Schema Design Apply the concepts learned to design the ERD of at least 2 basic and different types of applications. Apply the concepts learned and convert the ERD designed in the previous lab into a relational schema.	04
Week 3 & 4	Introduction to SQL and Database Creation Introduction to SQL and database software (e.g., MySQL) Creating databases and tables Inserting data into tables Retrieving data using SELECT statements	04
Week 5 & 6	Basic Data Retrieval and Filtering Using WHERE clause for filtering data Sorting and ordering results Updating and deleting records	04

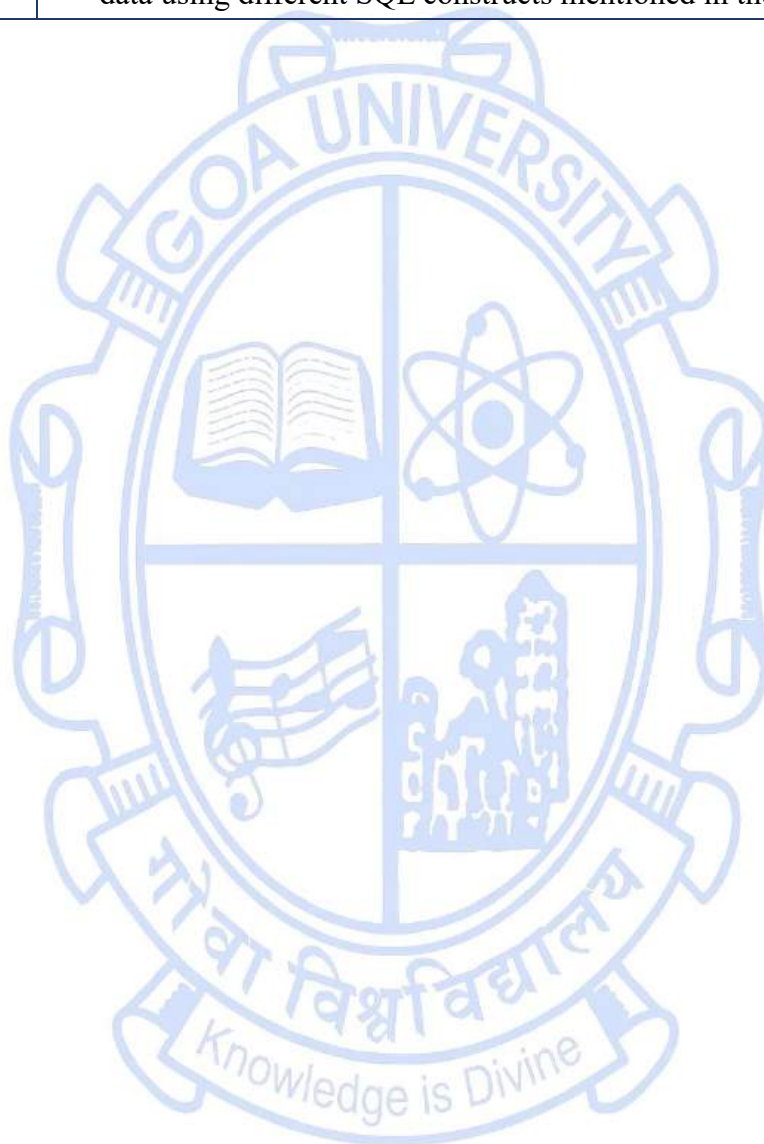
Week 7 & 8	Data Manipulation and Constraints Altering tables, column(s) Applying constraints (Primary Key, Foreign Key, Unique, Not Null, Check)	04
Week 9 & 10	Advanced Querying and Joins - Writing queries with Inner Join, Left Join, and Right Join - Using GROUP BY, ORDER BY and HAVING for grouping and filtering data	04
Week 11 & 12	Subqueries and Nested Queries - Writing subqueries within SELECT, INSERT, and UPDATE statements - Using nested queries for complex data retrieval	04
Week 13	Apply the concepts learnt to show the step-wise normalization process of tables from 1NF till 3NF by outlining appropriate reasoning of at least 2 basic types of applications.	04
Week 14 & 15	Views, Triggers, Stored Procedures Write and use simple Views, Triggers & Stored procedures	02
Assignment Mini-Project	- Students create a small database project related to real-life applications, such as library management, e-commerce, or student records. - Project includes database design, table creation, and implementation of queries and constraints.	
Pedagogy:	Apart from the normal teach-learning methodology, following are suggested. - Video /animation to explain various concepts - Collaborative, peer, flipped learning etc. - Adopt problem-based learning which foster student's analytical skills and develop design thinking skills.	
References/ Readings:	Text Book - Silberschatz, A., Korth, H., & Sudarshan, S. (2013) <i>Database system concepts</i> (6th ed.). McGraw-Hill. Reference Books - Date, C. J., Kannan, A., & Swaminadhan, S. (2006) <i>An introduction to database systems</i> (8th ed.). Pearson. - Elmasri, R., & Navathe, S. B. (2015) <i>Fundamentals of database systems</i> (7th ed.). Pearson Education. - Ramakrishnan, R., & Gehrke, J. (2002) <i>Database management systems</i> (6th ed.). McGraw-Hill . - Connolly, T., & Begg, C. (2015) <i>Database systems: Concepts, design and applications</i> (6th ed.). Pearson. - Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008) <i>Database systems: The complete book</i> (2nd ed.). Pearson.	

**Course
Outcomes:**

On completion of the course, students will be able to

1. Remember the basic concepts and terminologies of DBMS, ERD, Relational Model, Normalization, and Transaction Processing.
2. Understand ER diagrams, Normalization, relational schema design, Transaction Processing, and SQL concepts.
3. Apply ER Diagram, Relational Model, Normalization, and SQL Concepts
4. Design relational database and formulate queries on the database and data using different SQL constructs mentioned in the syllabus.

[\(Back to Index\)](#)



Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-121
Title of the Course	: SSC/Q0501 Software Developer
Number of Credits	: 12
Effective from AY	: 2025-26

Pre-requisites for the Course:	Basic understanding of Software Development and Programming Languages	
Course Objectives:	1. Development of software applications and interfaces as well as enhancements to existing packaged applications or pre-engineered templates and providing support to custom applications, debugging, maintenance and documentation. 2. As mentioned in the following NOS a. SSC/N0501: Contribute to the design of software products and applications b. SSC/N0502: Develop software code to specification c. SSC/N9014: Maintain an inclusive, environmentally sustainable workplace. d. DGT/VSQ/N0102: Employability Skills	
Units	Content	No of hours 423
Unit 1	SSC/N0501: Contribute to the design of software products and applications • IT-ITeS/IT Services Industry – An Introduction (Bridge Module) • Introduction to IT-ITeS industry and its Various Sub-Sectors • Career Path for a Software Developer • IT Services – An Introduction (Bridge Module) - Software development & IT Services Module 1: Programming and Algorithms - Software Programming & Algorithm Development Module 2: Analysis and Design of Software Applications - Software Development Life Cycle - Elements of Software Development Process Module 3: Work requirement, tools, and software - Program Specification and Programming Language - Approaches to Develop Applications and the Key Processes	33T + 30P
Unit 2	SSC/N0502 Develop software code to specification Module 4: Process of Software Development - Software Development Lifecycle	54 T + 48P

	Module 5: Process of Application Development - Quality Attributes of Software Requirements Specification - Custom and Rapid Application Software Module 6: Concept of Software Testing - Software Testing	
Unit 3	SSC/N9014 Implement & Improve the Gender Sensitivity, PWD (Person/People with Disability) Sensitivity and Greening Module 7: Inclusive and environmentally sustainable workplaces - Sustainable Practices - Respect Diversity and Strengthen Practices to Promote Equality	10 T + 10 P
Unit 4	Employability Skills Module 8: Introduction to Employability Skills Module 9: Constitutional values - Citizenship Module 10: Becoming a Professional in the 21st Century Module 11: Basic English Skills Module 12: Career Development & Goal Setting Module 13: Communication Skills Module 14: Diversity & Inclusion Module 15: Financial and Legal Literacy Module 16: Essential Digital Skills Module 17: Entrepreneurship Module 18: Customer Service Module 19: Getting ready for apprenticeship & Jobs	24 T + 18 P
Pedagogy:	Apart from the normal teach-learning methodology and methodology suggested in QP, the following are suggested. 1. Video /animation to explain various concepts 2. Collaborative, peer, flipped learning etc. 3. Adopt problem-based learning which foster student's analytical skills and develop design thinking skills. 4. MOOCs can be recommended for more in depth and advanced understanding of the topics.	
References/ Readings:	1. IT-ITeS Sector Skills Council NASSCOM. (2024, June). Participants Handbook – Software Developer. 2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms (3rd ed.). MIT Press. 3. Elmasri, R., & Navathe, S. B. (2010). Fundamentals of database systems (6th ed.). Pearson Education.	

	- Schwalbe, K. (2017). An introduction to project management (6th ed.). Course Technology. - Larman, C. (2004). Agile and iterative development: A manager's guide. Addison-Wesley. - van Vliet, H. (2008). Software engineering: Principles and practice (3rd ed.). Wiley.
Course Outcomes:	- Implement appropriate standards to assist in performing software construction as per specifications - Identify software development needs and changes. - Design algorithms to solve problems and execute test cases to convert them into code. - Evaluate the various software testing methodology and identify the correct one to deploy. - Analyze software designs for already built products. - Build database skills including DBMS, data design for pre development process. - Categorize between UML and Object-Oriented Design. - Discuss about manual and automated testing of software components. - Demonstrate application of suitable Unit Test Cases to validate the process of testing. - Demonstrate effective communication and collaboration with colleagues. - Apply measures to maintain standards of health and safety at the workplace. - Use different approaches to effectively manage and share data and information. - Develop strong relationships at the workplace through effective communication and conflict management. - Identify best practices to maintain an inclusive, environmentally sustainable workplace.
Qualification Pack & Courseware web links	Qualification Pack : https://pursuite-production.s3.ap-southeast-1.amazonaws.com/Final_Courseware/Final_QP/Version_3/QP_SS_C_Q0501_v3.0.pdf Courseware: https://pursuite-production.s3.ap-southeast-1.amazonaws.com/Final_Courseware/Updated+PH/English/PH_English_Software_Developer_IT%20Services_SSC_Q0501_V3.0.pdf

Project (on/off Campus)

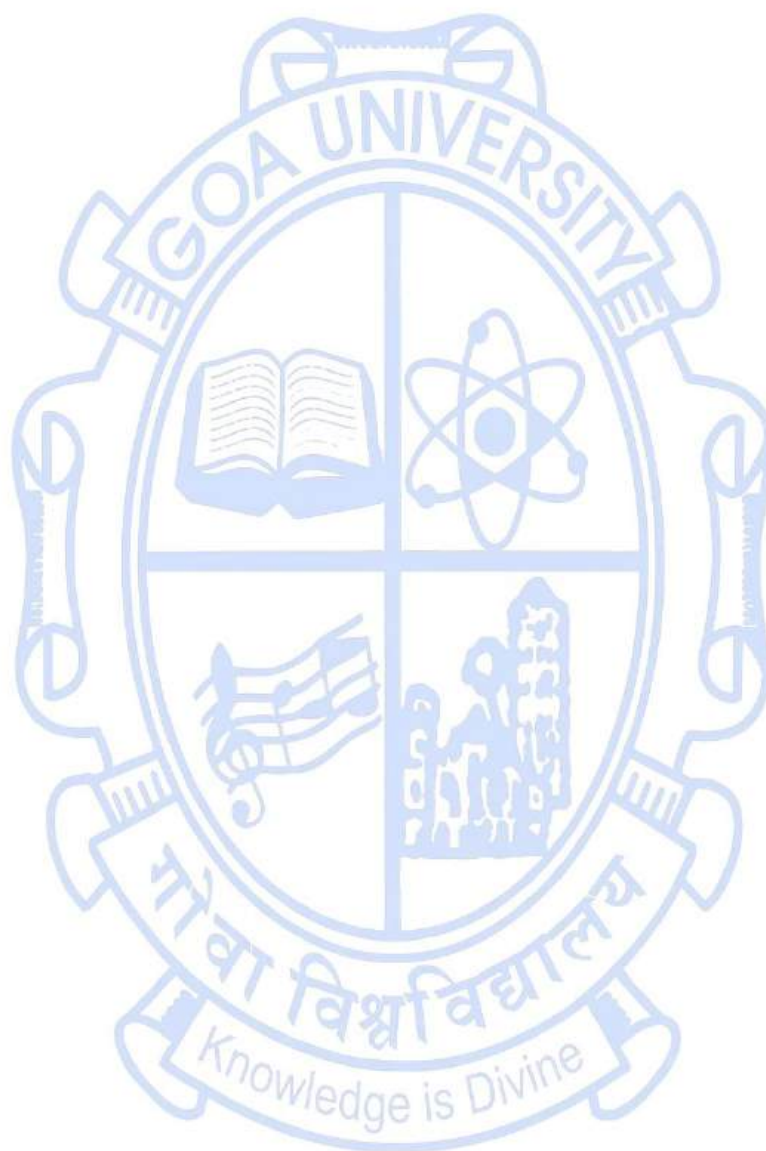
Minimum Duration: 90 Hours

It is required by a student to carry-out and complete

- On/ Off-Campus Project in the relevant skill area (Software Developer).
- Project can be done individually or in a group of maximum 5 students.

- There shall be an internal project guide and external, if required.
- After completion of the project, all the artifacts along with the project report in relevant format to be submitted in the college for evaluation and feedback.

[\(Back to Index\)](#)



SEMESTER-II

Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-102
Title of the Course	: Object-Oriented Concepts using Java
Number of Credits	: 3 (2T + 1P)
Effective from AY	: 2025-26

Prerequisites for the Course:	Knowledge of any basic Programming Language	
Course Objectives:	1. To remember Object-Oriented Programming concepts. 2. Become acquainted with the concepts of java & objected orientation: abstraction, encapsulation, inheritance, polymorphism, and apply them in problem-solving 3. Explain the concept of exception handling, multithreading, 4. To apply Object Oriented concepts to solve real-world problems.	
Units	Content	No of hours 45 (30T+15P)
I	Introduction to OO Programming Introduction to Object-oriented programming Problems/ Limitations of Procedure-Oriented Programming Comparison of Procedure-Oriented and Object-Oriented Paradigms Object Oriented Programming Paradigms i. Classes & Objects ii. Inheritance iii. Polymorphism iv. Abstraction v. Encapsulation Variables, scope, methods and Class Diagram Introduction to variables, scope of variables-local, instance and class variables, Objects, Class, attributes, methods, static methods Relationship between Classes/ Objects using class diagrams and Aggregation Constructors and Polymorphism Introduction, Types of Constructors, Compile and run time polymorphism Operator and Function Overloading Introduction Examples	15

II	Inheritance Introduction, Base class and derived classes Private, Public and Protected members Types of Inheritance - Single Inheritance - Multilevel Inheritance - Multiple Inheritance - Hierarchical Inheritance - Hybrid Inheritance Method overriding Virtual base classes (concept only) Abstract classes and Interfaces Exception Handling Introduction Types of errors Exception types-checked and unchecked Exception Handling Mechanism: Using try catch and multiple catch Nested try, throw, throws, and finally Creating user-defined Exceptions	15
III	Practical Work The use of JAVA programming language for the concepts learned in the units from I and II is required to be implemented practically. The broad area of practical Problems is mentioned below.	Practical Hours (30)
Week 1 to 3	Application/Use of language, Simple Programs, arithmetic, logical and relational operators, Data types, Control statements, and Java Packages (Scanner, math), break and continue in loops. Predefined Java String and math functions	06
Week 4 & 5	Implementing Classes , objects and Initialization	04
Week 6 to 8	Reading and writing data using methods, Modes of Parameter passing, and Return keyword.	06
Week 9 & 10	Programs on Constructors: Default, Parameterized, and Copy constructor	04
Week 11 & 12	Programs on Polymorphism: Function Overloading and function overriding, super keyword	04
Week 13 to 15	Programs on Exception Handling in Java	06
Pedagogy:	Apart from the normal teach-learning methodology, following are suggested. 1. Video /animation to explain various concepts 2. Collaborative, peer, flipped learning etc. 3. Adopt problem-based learning which foster student's analytical skills and develop design thinking skills.	

References/ Readings:	Text Books: 1. Bhawe, M., & Patekar, S. (2008). <i>Programming with Java</i> (1st ed.). Pearson. 2. Balagurusamy, E. (2010). <i>Object-oriented programming with Java</i> (4th ed.). Tata McGraw Hill Publishing House. Reference Books: 1. Schildt, H. (2017). <i>The complete reference JAVA2</i> (10th ed.). Tata McGraw Hill Publishing House. 2. Bloch, J. (2018). <i>Effective Java</i> (3rd ed.). Addison-Wesley Professional. 3. Goetz, B., Peierls, T., Bloch, J., Bowbeer, J., Holmes, D., & Lea, D. (2006). <i>Java concurrency in practice</i>. Addison-Wesley Professional. 4. Horstmann, C. S., & Cornell, G. (2012). <i>Core Java Volume I—Fundamentals</i> (10th ed.). Prentice Hall. 5. Sierra, K., & Bates, B. (2005). <i>Head first Java</i> (2nd ed.). O'Reilly Media.
Course Outcomes:	On completion of the course, students will be able to: 1. Remember Object-Oriented Programming concepts. 2. Understand object-oriented paradigms: abstraction, encapsulation, inheritance, polymorphism, and apply them in problem-solving 3. Apply object-oriented solutions for real-world problems. 4. Implement appropriate object-oriented concepts in applications.

[\(Back to Index\)](#)

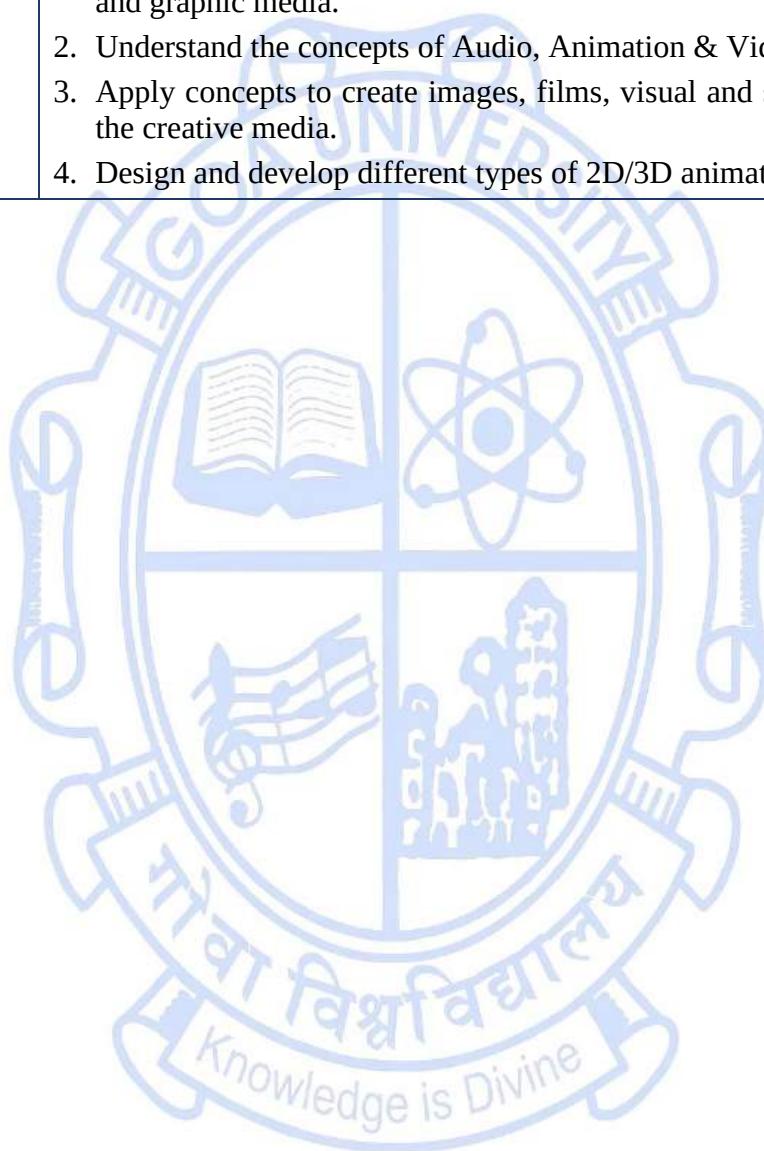
Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-103
Title of the Course	: Multimedia Technologies
Number of Credits	: 3(2T + 1P)
Effective from AY	: 2025-2026

Pre-requisites for the Course:	Basic Knowledge of Computers and Internet	
Course Objectives:	1. To understand color theory and color models. 2. To understand raster and vector graphic format and basic graphic editing 3. To understand basic audio and video formats, codecs, basic editing and filters. 4. Create and manipulate animations.	
Units	Content:	No of Hours 60 (30T+30P)
Unit I	Introduction: Introduction to Multimedia, Digital Representations, Social & Ethical considerations Color Theory: Color Basics, Color Models, Color Wheel, Color Relationships, Color Systems, Color Psychology, Shades, Tones & Tints Raster Graphics: Resolution, File Formats, Manipulation, Geometric transformations Vector Graphics: Fundamentals, File Formats, Shapes and Transforms Difference between Raster and Vector Graphics Text & Layout: Character Set, Fonts, Layout & Text in Graphics	15
Unit II	Sound: Sampling, Quantization, Audio Codec & File Formats, Processing Sound, Sound Editing and Effects, Compression, MIDI Audio Video: Basics of Video, Aspect Ratio, Frame Size, Frame Rate, Video Codec & File Formats, Processing & Delivery of Video Animation: Principles of Animation, Perception of Vision, Types of Animation, Keyframe, Techniques of animation, File Formats, Animated Gifs	15
Unit III	Practical Work Practical can be done using GIMP, Inkscape, Scribus, Photoshop, Illustrator, Opentoonz, Flash, Blender, Audacity, Lightworks	30

Week 1	Graphic Design: Image Composting: Remove background and combine images to create a work of art	2
Week 2 & 3	Graphic Design: Design a logo for a company	4
Week 4 & 5	Graphic Design: Design a brochure/poster for marketing a product	4
Week 6 & 7	Audio Editing: Audio recording, Audio storage and conversion Audio mixing and rendering.	4
Week 8 to 11	Video Editing: Video Capturing and Editing, Effects and transitions,color correction, Video composition and rendering. Prepare a video content with title and special effects	8
Week 12 to 15	Animation:- Creating simple frame by frame animation for a short animation, demonstrating the concept of layering and onion skinning (maximum 20sec with color drawings and background.) Ball animation: Drawing the ball with gradient color, Creating key frames for the animation sequence, Creating stretch and squash for the ball animation, Giving tween to the sequence of ball animation by connecting to path, duplicating waypoints, work with background image in the developed scene	8
Pedagogy:	Apart from the normal teach-learning methodology, following are suggested. 1. Video /animation to explain various concepts 2. Collaborative, peer, flipped learning etc. 3. Adopt problem-based learning which foster student's analytical skills and develop design thinking skills.	
References/ Readings:	Text Books: 1. Chapman, N., & Chapman, J. (2004) Digital multimedia (2nd ed., Wiley India Edition). Wiley. 2. Tay, V. (2011) Multimedia: Making it work (8th ed.). Tata McGraw-Hill. 3. Parekh, R.(2017) Principles of multimedia (2nd ed.). McGraw Hill Education. Reference Books: 1. Parker, (1997) R. One-minute designer (2nd ed.). Hungry Minds Inc. 2. Adobe Creative Team (2022) Adobe Photoshop classroom in a book. Adobe Press. 3. Adobe Creative Team (2024) Adobe Illustrator classroom in a book. Adobe Press. 4. Adobe Creative Team (2012) <i>Adobe Flash Professional CS6 classroom in a book</i> (1st ed.). Adobe Press. 5. Li, Z., & Drew, M. S. (2003) Fundamentals of multimedia (Pearson	

	Education International ed.). Pearson Education. 6. Jeffcoate, J. (1995) Multimedia in practice: Technology and applications. PHI. 7. Rahman, S. M. (2008) Multimedia technologies: Concepts, methodologies, <i>tools, and applications</i>. Minnesota State University, Mankato.
Course Outcomes:	On Completion of the course, students will be able to: 1. Remember the fundamentals and underlying theories of color theory and graphic media. 2. Understand the concepts of Audio, Animation & Video. 3. Apply concepts to create images, films, visual and sound effects for the creative media. 4. Design and develop different types of 2D/3D animations.

[\(Back to Index\)](#)



Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-104
Title of the Course	: Quantitative Techniques
Number of Credits	: 2T
Effective from AY	: 2 0 2 5 - 2 6

Pre-requisites for the Course:	NIL	
Course Objectives:	To equip students with foundational quantitative and analytical skills in arithmetic, financial mathematics, and basic statistics, enabling them to solve practical problems, interpret data, and make informed decisions in academic and real-world contexts.	
Units	Content:	No of hours
Unit 1	Number Systems and Basic Arithmetic Number System: Properties of integers, divisibility rules, prime numbers, least common multiple (LCM), greatest common divisor (GCD). Progressions: Arithmetic Progression (AP), Geometric Progression (GP), and its applications. Averages: Concept and application of averages, weighted averages. Allegations: Mixture and allegation problems. Percentage: Basic percentage calculations, percentage increase/decrease. Ratio & Proportion: Concept, direct and inverse proportion, applications in practical problems.	10
Unit II	Financial Mathematics and Work-Time Problems Profit & Loss: Cost price, selling price, mark-up, profit percentage, loss percentage. Discount: Trade discount, cash discount, successive discounts. Simple and Compound Interest: Basic concepts, differences, calculation of amount, rate, time, and principal. Time and Work: Work done, efficiency, pipes and cisterns, combined work. Time and Distance: Speed, distance, time relations, problems on trains, boats, and streams. Population Growth and Depreciation: Growth rate, depreciation formulae, calculation of population or asset value over time.	10
Unit III	Basic of Statistics Introduction to Statistics: Definition, types of data (qualitative	10

	and quantitative), data collection methods. Scales of Measurement: Nominal, Ordinal, Interval and Ratio Measures of Central Tendency: Mean, median, mode - calculation and applications. Measures of Dispersion: Range, variance, standard deviation, applications in data analysis. Correlation and Regression: Basics of correlation, scatter diagrams, introduction to linear regression.	
Pedagogy:	Apart from the normal teach-learning methodology, following are suggested. 1. Video /animation to explain various concepts 2. Collaborative, peer, flipped learning etc. 3. Adopt problem-based learning which foster student's analytical skills and develop design thinking skills.	
References/ Readings:	1. Verma, R. (2017). <i>Fast track objective arithmetic</i>. Arihant Publications Limited. 2. Gupta, S. C. (2018). <i>Fundamentals of statistics</i> (7th ed.). Himalaya Publishing House. 3. Sharma, A. (2021). <i>How to prepare for quantitative aptitude for CAT</i> (9th ed.). McGraw Hill. 4. Mishra, P. K., & Mishra, R. (2018). <i>Elementary & advanced mathematics for competitive exams</i>. Source Books. 5. Aggarwal, R. S. (2017). <i>Quantitative aptitude for competitive examinations</i>. S. Chand Publications. 6. Gupta, S. C., & Kapoor, V. K. (2020). <i>Fundamentals of mathematical statistics</i> (12th ed.). S. Chand and Sons.	
Course Outcomes:	On Completion of the course, students will be able to: 1. Understand and apply core principles of number systems, divisibility, LCM, GCD, and arithmetic progressions, enabling accurate and efficient problem-solving. 2. Perform calculations involving profit, loss, discounts, and interest, equipping them with practical skills to analyse and solve financial problems relevant to business and personal finance. 3. Solve real-world problems involving ratios, proportions, time-distance, and work efficiency, enhancing their quantitative reasoning and analytical abilities for everyday and professional applications. 4. Summarize, interpret, and draw insights from data using measures of central tendency, dispersion, correlation, and regression, supporting data-driven decisions.	

[\(Back to Index\)](#)

Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-122
Title of the Course	: Sr. Associate - Desktop Publishing (DTP)
Number of Credits	: 12
Effective from AY	: 2025-26

Pre-requisites for the Course:	To take the Desktop Publishing course, students should have basic computer and word processing skills, a creative mindset, and an interest in design. Knowledge of graphic design principles is helpful, and internet access is required for using tools and resources.	
Course Objectives:	1. Publishing content in digital, multimedia, web-based or printed formats. 2. Knowledge and understanding of Designing & Publishing Software 3. Granting access to publications in your organization's knowledge base only to authorized users, and ensuring the correct versions are used. 4. Promoting diversity, equality, and inclusion in a sustainable, eco-friendly workplace. 5. Developing key skills: employability, digital, financial, and legal literacy, and communication. 6. Promoting diversity, inclusion, and constitutional values. 7. Fostering career readiness through customer service, entrepreneurship, and professional growth.	
UNITS	Content:	No of hours 420
UNIT 1	Module 1: Introduction to the Job Role and Industry (Bridge Module)	36 T + 54 P
	• Job Role and Responsibility of a Sr. Associate DTP • An Introduction to the ITeS Sector • DTP Industry in India	
	Module 2: Concept of Publications (SSC/N2702)	
	• Various Nature of Publications • Page Designing of Various Types	
	Module 3: Provide/Control Access to Publications (SSC/N2702)	
UNIT 2	• Store Publications	54 T + 96 P
	Module 4: Technical Skills for Publishing Content (SSC/N2703)	
UNIT 2	• Various Tools of DTP	54 T + 96 P

	Module 5: Publish Content in Standard Formats (SSC/N2703)	
	• Quality Issues • Liaise with Production Teams	
	Module 6: Software Requirement for Publishing Content (SSC/N2703)	
	• Adobe InDesign • Adobe PageMaker • Adobe Photoshop • Corel Draw • MS Publisher	
	Module 7: Production Requirement for Publication Process (SSC/N2703)	
	• Production Process Used for Publication	
UNIT 3	Module 8: Maintain an Inclusive, Environmentally Sustainable Workplace (SSC/N9014)	10 T + 20 P
	• Sustainable Practices • Respect Diversity and Strengthen Practices to Promote Equality	
UNIT 4	Module 9: Employability Skills	24 T + 36 P
	• Identify employability skills required for jobs in various industries. • Identify and explore learning and employability portals. • Recognize the significance of constitutional values, including civic rights and duties, citizenship, responsibility towards society etc. and personal values and ethics such as honesty, integrity, caring and respecting others, etc. • Follow environmentally sustainable practices • Recognize the significance of 21st Century Skills for employment. • Practice the 21st Century Skills such as Self-Awareness, Behaviour Skills, time management, critical and adaptive thinking, problem-solving, creative thinking, social and cultural awareness, emotional awareness, learning to learn for continuous learning etc. in personal and professional life. • Use basic English for everyday conversation in different contexts, in person and over the telephone. • Read and understand routine information, notes, instructions, mails, letters etc. written in English.	

	● Write short messages, notes, letters, e-mails etc. in English. ● Understand the difference between job and career. ● Prepare a career development plan with short- and long-term goals, based on aptitude. ● Follow verbal and non-verbal communication etiquette and active listening techniques in various settings. ● Work collaboratively with others in a team. ● Communicate and behave appropriately with all genders and PwD. ● Escalate any issues related to sexual harassment at workplace according to POSH Act. ● Select financial institutions, products and services as per requirement carry out offline and online financial transactions, safely and securely identify common components of salary and compute income, expenses, taxes, investments etc. ● Identify relevant rights and laws and use legal aids to fight against legal exploitation ● Operate digital devices and carry out basic internet operations securely and safely. ● Use e- mail and social media platforms and virtual collaboration tools to work effectively. ● Use basic features of word processor, spreadsheets, and presentations. ● Identify different types of Entrepreneurship and Enterprises and assess opportunities for potential business through research. ● Develop a business plan and a work model, considering the 4Ps of Marketing Product, Price, Place and Promotion. ● Identify sources of funding, anticipate, and mitigate any financial/ legal hurdles for the potential business opportunity. ● Identify different types of customers. ● Identify and respond to customer requests and needs in a professional manner. ● Follow appropriate hygiene and grooming standards ● Create a professional Curriculum vitae (Résumé). ● Search for suitable jobs using reliable offline and online sources such as Employment exchange, recruitment agencies, newspapers etc. and job portals, respectively. ● Apply to identified job openings using offline /online methods as per requirement . ● Answer questions politely, with clarity and confidence, during recruitment and selection.	
--	---	--

	Identify apprenticeship opportunities and register for it as per guidelines and requirements	
References/ Readings:	- IT-ITeS Sector Skills Council NASSCOM. (2024, June). <i>Participants Handbook – Associate DTP</i>. - Adams, S., Morioka, N., & Stone, T. L. (2018). <i>Color design workbook: A real-world guide to using color in graphic design</i> (1st ed.). Rockport Publishers. - Drew, J. T., & Meyer, S. A. (2012). <i>Color management: A comprehensive guide for graphic designers</i> (1st ed.). Allworth Press. - Faulkner, A., & Chavez, C. (2018). <i>Adobe Photoshop CC classroom in a book</i> (1st ed.). Pearson Education. - Bouton, G. D. (2014). <i>CorelDRAW X7: The official guide</i> (11th ed.). McGraw-Hill Education. - Carter, D. E. (2009). <i>The big book of layouts</i>. Harper Design. - Tondreau, B. (2019). <i>Layout essentials: 100 design principles for using grids</i>. Rockport Publishers. - Ambrose, G., & Harris, P. (2008). <i>The production manual: A graphic design handbook</i>. AVA Publishing. - Samara, T. (2007). <i>Design elements: A graphic style manual</i>. Rockport Publishers	
Course Outcomes:	- Students will have competencies and skills as mentioned in the model curriculum of Associate - DTP designed by NASSCOM - Students will be job ready to start their career in the DTP field - This course will help students produce polished, high-quality documents for academic or career-related purposes. - Students gain proficiency in software tools like Adobe InDesign, Microsoft Publisher, and others, boosting their digital literacy.	
Qualification Pack & Courseware web links	Qualification Pack : https://pursuite-production.s3.ap-southeast-1.amazonaws.com/Final_Courseware/Final_QP/Version_3/QP_SS_C_Q2702_v3.0.pdf Courseware: https://s3.ap-south-1.amazonaws.com/insdms2.0/Standard%20Management/Sr.%20Associate-Desktop%20Publishing%20%28DTP%29_Version%203.0/Standards%20PH/PH_English_Sr.%20Associate%20-%20Desktop%20Publishing%20%28DTP%29_SSC_Q2702_V3.0.pdf	

Project (on/off Campus) / On-Job-Training (OJT)/Internship

Total Duration: 90 Hours i.e. equivalent to 1.5 Months to 2 Months.

It is required by a student to carry-out and complete

On/Off-Campus Project in the relevant skill area (Desktop Publishing).

- Project can be done individually or in group of maximum 5 students.
- There shall be an internal project guide and external, if required.
- After completion of the project, all the artifacts along with project report in relevant format to be submitted in the college for evaluation and feedback.

OR

OJT in the relevant industry and in the relevant skill area (Desktop Publishing).

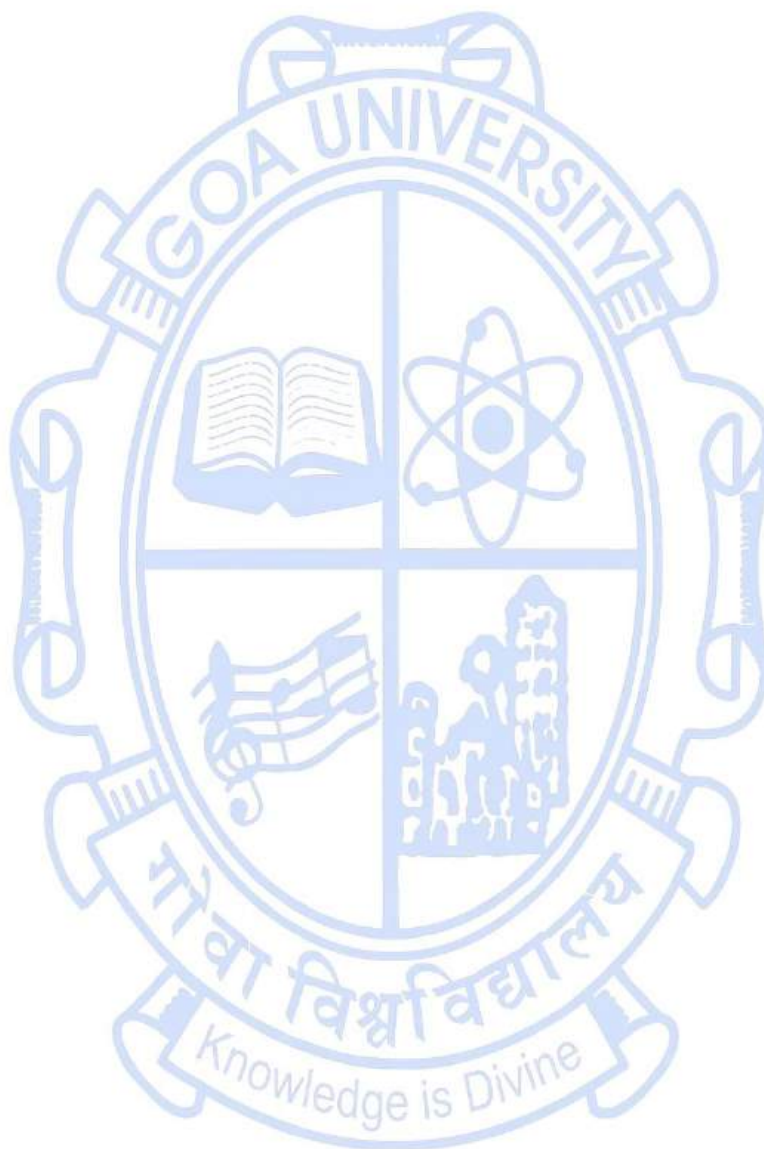
- A Student needs to submit the detailed report of the OJT along with the 'Certificate of Completion' for evaluation and feedback.

OR

Internship in the relevant industry and in the relevant skill area (Desktop Publishing).

- A Student needs to submit the detailed report and work diary along with the 'Certificate of Completion' for evaluation and feedback.

[\(Back to Index\)](#)



EXIT MODULE

Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-161
Title of the Course	: Project Development
Number of Credits	: 4 (2T + 2P)
Effective from AY	: 2025-26

Pre-requisites for the Course:	Knowledge of Programming and Designing	
Course Objectives:	1. Comprehend the structured lifecycle and methodologies involved in software project development. 2. Apply user-centered design principles for effective interface planning and usability enhancement. 3. Develop proficiency in requirement analysis, system design, and technical documentation practices. 4. Design and build a functional software prototype with minimal coding, emphasizing planning and design accuracy.	
Units	Content	No. of Hours 90 (30T + 60P)
I	<i>The emphasis of this course is more on Design, Planning, Documentation, Prototyping</i> Introduction to Software Design & Project Planning 1. Software Project Basics • What is a Software Project? • Software Development Life Cycle (basic understanding) • Importance of design before coding • Real-life examples of successful software products 2. Design Thinking in Software • Introduction to Design Thinking • Empathy Mapping • Identifying User Needs • Defining Problem Statements 3. Requirement Analysis • Functional and Non-functional Requirements • User Stories (basic format) • Preparing a simplified SRS document	15

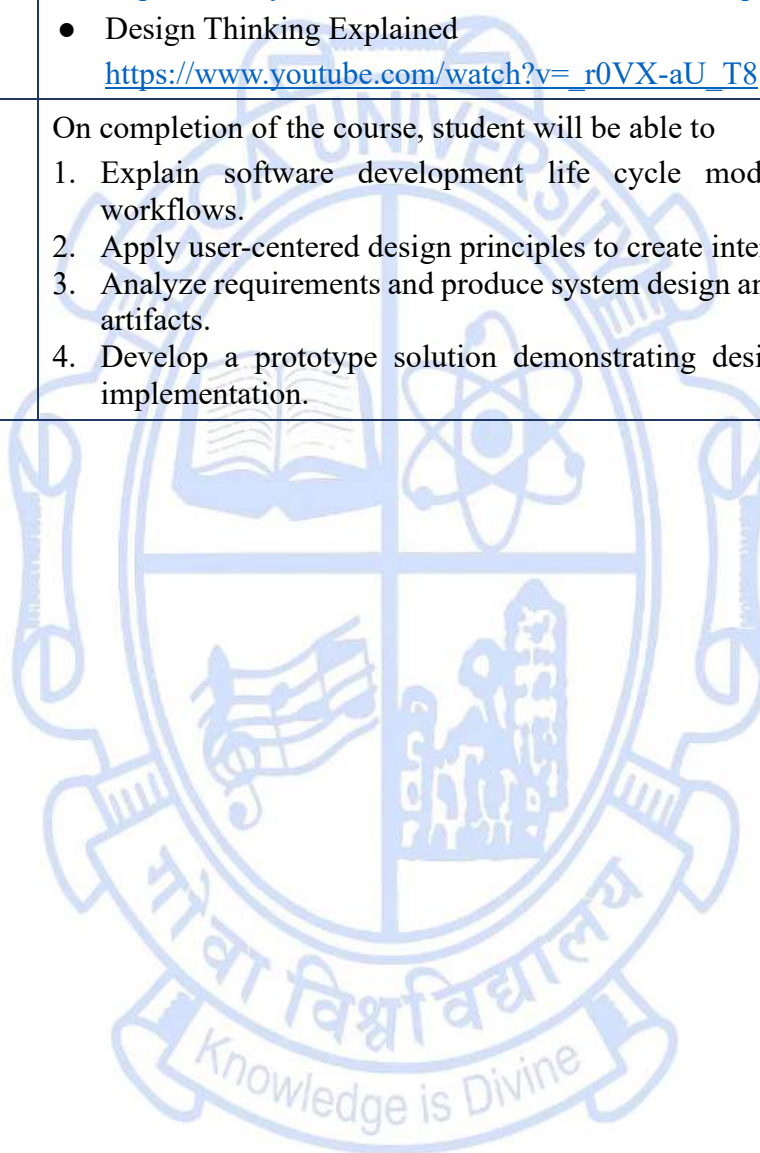
II	System Design & User Experience 1. Information Architecture • Site maps • Navigation structure • Content organization 2. Wireframing & Prototyping • Low-fidelity wireframes (paper/digital) • High-fidelity mockups (Figma/Canva/Adobe XD) • Design consistency and visual hierarchy • Typography and color principles 3. Basic System Modeling • Use Case Diagrams • Flowcharts • Basic Database Schema Design (concept only) 4. Introduction to Minimal Implementation • Converting prototype to simple functional screens • Basic form handling (optional) • UI consistency testing	15
III	Practical Activities - To be carried out alongside Unit I & II. 1. Identify a real-world problem and prepare a Design Brief. 2. Create User Personas and Empathy Maps. 3. Prepare: • Problem Statement • User Stories • SRS (simplified format) 4. Develop: • Sitemap • Navigation Structure • Flowcharts 5. Design: • Low-fidelity wireframes (at least 5 screens) • High-fidelity prototype 6. Conduct basic usability testing (peer review). 7. Improve prototype based on feedback. 8. Develop minimal functional demo (optional – basic linking between screens).	30

	9. Prepare complete documentation and presentation. Suggested Project Types: • Event Management App (Prototype) • College Helpdesk Portal • Food Delivery App UI • Student Planner Application • Tourism Information Portal • NGO Donation Website Prototype	
IV	Case Studies (Design-Centered Analysis) 1. UI evolution of a popular mobile app 2. Redesign of a poorly designed website 3. User experience analysis of an e-commerce platform <i>Students will analyze:</i> • User Interface • User Experience • Navigation • Accessibility • Improvements Mini Project (Design Prototype-Based) 1. Individual / Team (2 students) 2. Develop a complete software design prototype 3. Submission includes: • Design Brief • Personas • SRS • Sitemap • Wireframes • High-Fidelity Prototype • Usability Testing Report • Presentation (Only minimal coding required — focus on design quality) <i>Optional: Students may prepare for industry-recognized certifications in UI/UX design, software development fundamentals, and design thinking by completing online courses, practice exercises, and certification programs offered by reputed platforms such as Google, Coursera, IBM, Meta, or LinkedIn Learning. These certifications help students develop practical skills in user experience design, interface prototyping, project management, and basic software</i>	30

	<i>development practices, thereby improving their industry readiness and employability.</i> <i>Some beginner-friendly certification options include:</i> • Google UX Design Professional Certificate – Coursera • Meta Front-End Developer Certificate – Coursera • Figma UI/UX Design Certification Courses – Udemy / LinkedIn Learning	
Pedagogy:	Suggested strategies for use to accelerate the attainment of the various course outcomes. 1. A plan is to be developed by the student/s in consultation with the teacher incharge and to be approved. 2. One or methods mentioned below may be used for learning purposes. ✓ Intensive training / teaching ✓ Online or offline training (approved by the college or instructor) ✓ Approved MOOCS Courses ✓ Workshops - on-campus or off-campus ✓ Self-learning means & methods ✓ Enquiry-based learning 3. A work diary to be maintained where all the learning & work carried out to maintained and certified by the teacher incharges. 4. All deliverable & artifacts to be submitted in the college for evaluation and assessments.	
References/ Readings:	Main Reading: 1. Krug, S. (2014). Don't make me think, revisited: A common sense approach to web usability (3rd ed.). New Riders. 2. Garrett, J. J. (2011). The elements of user experience: User-centered design for the web and beyond (2nd ed.). New Riders. 3. Duckett, J. (2022). HTML and CSS: Design and build websites (Updated ed.). John Wiley & Sons. 4. Gothelf, J., & Seiden, J. (2021). <i>Lean UX: Designing great products with agile teams</i> (3rd ed.). O'Reilly Media. Additional Reading/Resources: 1. Norman, D. A. (2013). <i>The design of everyday things</i> (Revised and expanded ed.). Basic Books. 2. Tidwell, J., Brewer, C., & Valencia, A. (2020). <i>Designing interfaces: Patterns for effective interaction design</i> (3rd ed.). O'Reilly Media. 3. Krug, S. (2009). Rocket surgery made easy: The do-it-yourself guide to finding and fixing usability problems. New Riders. 4. Babich, N., & Smart, C. (2023). <i>UX design for beginners: A crash course in 100 short lessons</i>. UX Planet. 5. Anderson, D. (2022). <i>Practical UI design with Figma</i>. Packt	

	Publishing. 6. Beginner Video Tutorials • UI/UX Design Full Course for Beginners https://www.youtube.com/watch?v=Ovj4hFsko7c • Software Development Life Cycle Explained https://www.youtube.com/watch?v=i-QyW8D3ei0 • Figma UI Design Tutorial for Beginners https://www.youtube.com/watch?v=FTFaQWZBqQ8 • Design Thinking Explained https://www.youtube.com/watch?v=_r0VX-aU_T8
Course Outcomes:	On completion of the course, student will be able to 1. Explain software development life cycle models and project workflows. 2. Apply user-centered design principles to create interface layouts. 3. Analyze requirements and produce system design and documentation artifacts. 4. Develop a prototype solution demonstrating design and minimal implementation.

[\(Back to Index\)](#)



SEMSTER III

Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-200
Title of the Course	: Web Designing Concepts
Number of Credits	: 3 (2T +1P)
Effective from AY	: 2026-27

Pre-requisites for the Course:	Nil	
Course Objectives:	1. To understand the basic web design principles, website planning process, and fundamental concepts involved in developing user-friendly websites. 2. To develop structured web pages using HTML/HTML5 elements, forms, multimedia features, and semantic tags. 3. To analyze and apply CSS/CSS3 concepts to design and style web pages using layouts, box model, positioning, navigation bars, image effects, and responsive design techniques. 4. To implement basic JavaScript programs to add interactivity to web pages using variables, control structures, functions, DOM manipulation, event handling, and form validation.	
Units		No of hours
I	Web Design Principles – Overview Brief History of Internet, what is World Wide Web, Why create a website. Basic principles involved in developing a web site, Web Standards by W3CProtocols, Planning process. Design Concept –learnability, visibility, error prevention, efficiency and graphic design.	5
II	Structure and Style with HTML and CSS A. HTML AND HTML5 Introduction, development process, basic HTML, formatting and fonts, commenting code, colors, hyperlinks, lists, tables, images, simple HTML forms, website structure, meta tags, character entities, Unicode fonts. HTML5: New elements, Canvas, SVG, drag and drop, geolocation, video, audio, input types, HTML5 forms (elements and attributes), semantic elements, web storage, application cache, web workers, and SSE. B. CSS and CSS3 Introduction – Syntax, inline, internal and external style, Id &	15

	Class, Backgrounds, Text, Fonts, Links, Lists, Tables. CSS Box Model – Border, Outline, Margin, Padding. Advanced - Grouping/Nesting, Dimension, Display, Positioning, Floating, Align, Pseudo-class, Pseudo element, Navigation Bar, Image Gallery, Image Opacity, Image Sprites, Media Types, Attribute Selectors. Gradients, Text Effects, Fonts, 2D Transforms, 3D Transforms, Transitions, Animations, Multiple Columns.	
III	Javascript Introduction to JavaScript – features, role in web development, adding JavaScript to HTML (internal and external), simple programs, and basic event handling. Core concepts – comments, variables (var, let and const), scope (global and local), data types, operators, conditional statements (if), loops (for, while), and functions. Introduction to objects and arrays, foreach loop. Basic DOM and Browser Object Model – window methods (alert(), confirm(), prompt()), getElementById(), and content manipulation using innerHTML and innerText. Basic client-side form validation including required field and email validation.	10
IV	Practical Work:	30
Week 1 & 2	Semantic HTML Structure - Create a "Digital Resume" using HTML and HTML 5.	04
Week 3 & 4	Introduction to Styling - Apply colors, Google Fonts, and external CSS Selectors, Box Model (margin, padding, border), and the <div> tag to the previously created resume.	04
Week 5 to 7	Audio, Video & animation - Create a "Gallery" with embedded Video/Audio. Animation - Create a simple animation using @keyframes and Transform Responsive Layouts & Multimedia - Build a multi-column "Magazine Layout" using Flexbox or CSS Grid.	06
Week 8 to 10	Interactivity – Ex1: Build a "Dark Mode" toggle Ex2: Create a "Simple Calculator."	06
Week 11 & 12	DOM Manipulation – Ex1: Capture user data and display it on a webpage. EX2: Use document.getElementById() to select the div and use .innerHTML to display a message.	04
Week 13 & 14	Validation - Create a “contact us” form and validate	04
Week 15	Create a multipage portfolio website.	02

Pedagogy:	Suggested strategies for use to accelerate the attainment of the various course outcomes are: 1. Students will be able to create structured web pages using HTML/HTML5 elements, forms, multimedia features, and semantic tags. 2. Students will learn to apply CSS/CSS3 techniques to design and format web pages using layouts, the box model, positioning, and various visual effects. 3. Students will learn to build interactive web pages using DOM manipulation, and form validation. 4. Assignment based learning.
References/ Readings:	Text Books 1. Robbins, J. N. (2018). <i>Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics</i> (6th ed.). O'Reilly Media. 2. Robson, E., & Freeman, E. (2014). <i>Head first HTML and CSS</i> (2nd ed.). O'Reilly Media. 3. Kogent Learning Solutions Inc. (2011). <i>HTML5 black book: Covers CSS3, JavaScript, XML, XHTML, Ajax, PHP and jQuery</i>. Pearson Education. Reference Books 1. Shneiderman, B., & Plaisant, C. (2010). <i>Designing the user interface: Strategies for effective human-computer interaction</i> (5th ed.). Pearson Education. 2. Kogent Learning Solutions Inc. (2011). <i>HTML5 in simple steps</i>. Dreamtech Press. 3. Bayross, I. (2015). <i>HTML5 and CSS3 made simple</i>. BPB Publications.
Course Outcomes:	At the end of this course, the students will be able to: 1. Understand the basic web design principles, website planning process, and fundamental concepts of developing user-friendly websites. 2. Create structured web pages using HTML/HTML5 elements, forms, multimedia features, and semantic tags. 3. Analyze and apply CSS/CSS3 techniques to design and format web pages using layouts, box model, positioning, and visual effects. 4. Develop interactive web pages using basic JavaScript concepts including variables, control structures, DOM manipulation, and form validation.

[\(Back to Index\)](#)

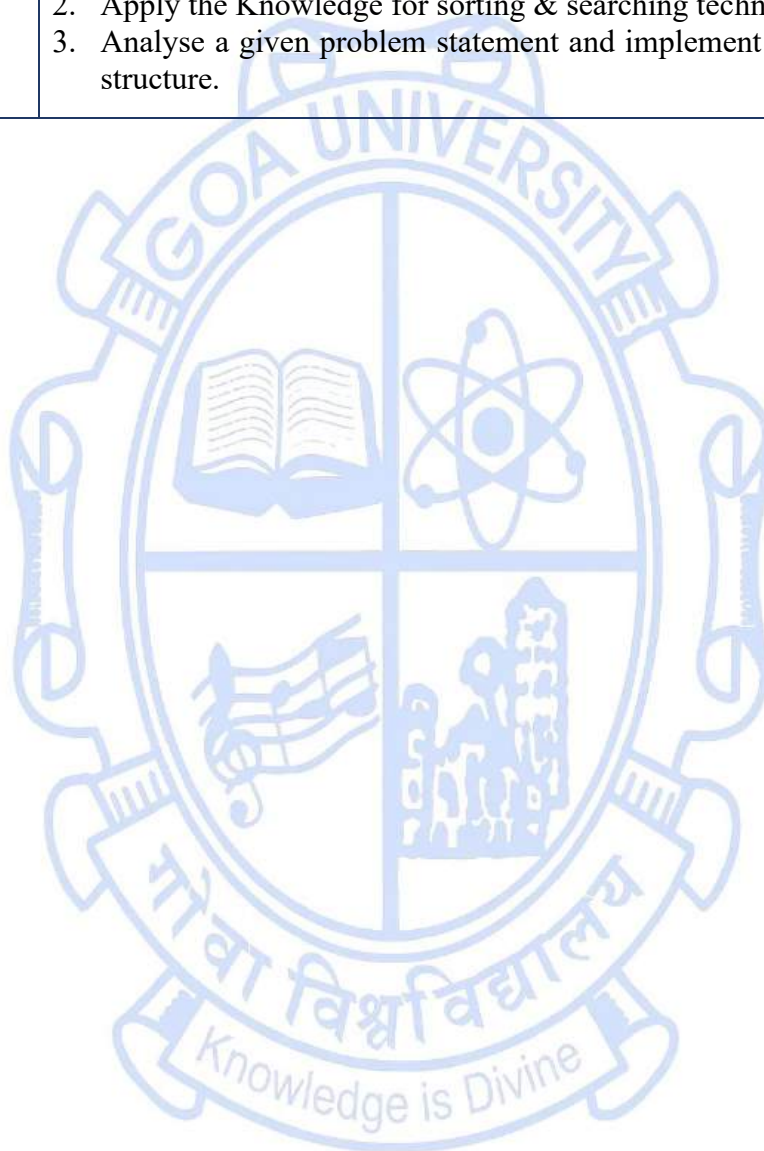
Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-201
Title of the Course	: Data Structures
Number of Credits	: 3 (2T +1P)
Effective from AY	: 2026-27

Pre-requisites for the Course:	Nil	
Course Objectives	1. To understand the concepts of Data Structures and Strings. 2. To apply Searching Techniques and Sorting Techniques. 3. To apply the use of Stack and Queue. 4. To understand the concepts of Trees and Graphs.	
Unit	Content	No. of Hours
I	Data Structure Introduction & Meaning, Data Type, Need for Data Structures, Types Strings Overview of Strings and string functions (strlen, strlen,strupr, strcpy, strcat, strcmp, strcmpi etc) Searching Techniques Linear Search & Binary Search Sorting Techniques Bubble, Selection, Insertion Introduction to Merge, Quick and Heap sort. Stack and Queue Concept, Representation, Operations Applications using stacks -Infix to postfix, Infix to prefix, Evaluation of expressions.	15
II	Linked Lists Concept, Representation & Basic Operations(insert, delete) Linked List Types - Doubly & Circular Trees Binary Trees, Binary Search Tree (BST)- Definition, Terminologies and applications, Traversals: Inorder, Preorder, Postorder. Operations on BST – Create, Insert, Delete, Search Applications: Expression Tree.	15

	Introduction to AVL, B and B+ Trees Graphs Concept & terminologies Graph Representation – Adjacency matrix, Adjacency list Traversals – BFS & DFS Applications of Graph - Shortest Path Algorithm	
III	Practical Work Using C programming language. Data Structure concepts to be covered as below:	30
Week 1	String Functions	02
Week 2 to 3	Searching Algorithms (Linear and Binary)	04
Week 4 to 6	Sorting Algorithms (Bubble, Selection and Insertion)	06
Week 7	Stack (Push & Pop operations)	02
Week 8	Queue (Enqueue & Dequeue operations)	02
Week 9 to 11	Singly Linked List (Implementation, traversal, addition and deletion of nodes)	06
Week 12 & 14	Binary Search Tree (Creation and traversal)	06
Week 15	Graphs - Adjacency matrix	02
Pedagogy:	Apart from the normal teaching-learning methodology, following are suggested. 1. Video /animation to explain various concepts 2. Adopt problem-based learning which fosters student's analytical skills and develop design thinking skills. 3. MOOC Courses and Scenario Based teaching for understanding different Data Structures 4. Flipped Learning Techniques in Teaching	
References/ Readings:	Recommended Text Books: 1. Balagurusamy, E. (2025). <i>Data structures using C</i> (2nd ed.). McGraw Hill Education. 2. Kanetkar, Y. (2024). <i>Data structures through C</i> (5th ed.). BPB Publications. 3. Thareja, R. (2018). <i>Data structures using C</i> (2nd ed.). Oxford University Press.	

	Recommended References: 1. Aho, A. V., Hopcroft, J. E., & Ullman, J. D. <i>Data structures and algorithms</i>. Pearson Education India. 2. Gupta, P., Agarwal, V., & Varshney, M. (n.d.). <i>Data structure using C</i> (2nd ed.). Laxmi Publications.
Course Outcome:	After completion of the course the student should be able to : 1. Understand the concepts of Data Structures as per the syllabus. 2. Apply the Knowledge for sorting & searching techniques. 3. Analyse a given problem statement and implement appropriate data structure.

[\(Back to Index\)](#)



Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-202
Title of the Course	: Reasoning Techniques
Number of Credits	: 2T
Effective from AY	: 2026-27

Pre-requisites for the Course:	None	
Course Objectives:	1. To understand the fundamental concepts of propositions, logical connectives, and argument structures in formal logic. 2. To apply verbal and non-verbal reasoning techniques to solve structured and unstructured problems. 3. To analyse statements, arguments, and reasoning patterns to determine validity and logical consistency. 4. To evaluate real-life situations and decision-making scenarios using appropriate reasoning and decision-making skills.	
Units		No of hours
I	STATEMENTS AND LOGIC Propositions and Statements, Logical Connectives, Compound Statements, Conjunction and Disjunction, Truth Tables, Logical Equivalence, Tautology, Contradiction and Contingency, Algebra of Propositions, Implications: Converse, Inverse, Contrapositive. Predicate and Quantifiers, Arguments and Types of Arguments, Validity of Arguments, Statements and Assumptions, Statements and Conclusions, Statements and Courses of Action, Syllogisms (Venn Diagram Method).	15
II	VERBAL AND NON-VERBAL REASONING Analogies, Classification and Odd One Out, Series Completion (Alphabet, Number, Mixed), Coding–Decoding, Simple and Coded Inequalities. Blood Relation, Direction Sense Problems, Ranking and Time Sequence Test, Problems on Seating Arrangements (Linear and Circular), Puzzles and pattern recognition. APPLIED REASONING AND DECISION MAKING Clocks and calendars, decision making problems, Judgement problems, situation reaction tests, data interpretation and data sufficiency problems.	15

[\(Back to Index\)](#)

Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-221
Title of the Course	: SSC/Q8404: User Experience Designer (Version 3.0)
Number of Credits	: 12 (5 Theory, 4 Practical, 3 Project)
Effective from AY	: 2026-27

Pre-requisites for the Course:	Conceptual knowledge of Graphic Design and Programming	
Course Objectives:	Objectives mentioned in the following National Occupational Standards (NOS) 1. SSC/N8453: Develop User Experience (UX) design guidelines and best practices. 2. SSC/N8420: Design and define style guide/ design system and specifications for developers. 3. SSC/N8454: Conduct Business Analysis and Develop User-Centric Use Cases. 4. SSC/N8422: Conduct a competitive analysis on strength and weakness of competitors products. 5. SSC/N8423: Create user personas to encapsulate and communicate user behaviour patterns. 6. SSC/N8424: Develop sitemap and information architecture for the solution planned for the customer. 7. SSC/N8425: Develop user experience maps, user journeys and user flows. 8. SSC/N8426: Develop detailed wireframes to illustrate flow, interactions and interface elements. 9. SSC/N8427: Develop a visual design / screens for the proposed wireframe. 10. SSC/N8428: Design the assets such as imagery, graphics and animations for the user interface. 11. SSC/N8429: Conduct usability tests on the planned user experience designs at different stages. 12. SSC/N8455: Design Accessible and Inclusive User Experiences. 13. DGT/VSQ/N0101: Employability Skills.	
UNITS	Content:	No. of Hours 285
Unit 1	Bridge Modules Module 1: IT-ITeS Industry – An Introduction Module 2: Web technology – An Introduction Module 3: Mobile Development	16 T & 18 P

	Module 4: Development Tools & Usage Module 5: User Experience Design Process (SSC/N8453: Develop User Experience (UX) design guidelines and best practices.) Module 6: Style guide, design systems and specifications (SSC/N8420: Design and define style guide/ design system and specifications for developers.)	
Unit 2	Module 7: Business requirements (SSC/N8454: Conduct Business Analysis and Develop User Centric Use Cases) Module 8: Competitive Analysis (SSC/N8422: Conduct a competitive analysis on strength and weakness of competitors products) Module 9: User persona and behavior patterns (SSC/N8423: Create user personas to encapsulate and communicate user behaviour patterns) Module 10: Site map and information architecture (SSC/N8424: Develop sitemap and information architecture for the solution planned for the customer) Module 11: User experience map, journey and user flow (SSC/N8425: Develop user experience maps, user journeys and user flows)	25 T & 38 P
Unit 3	Module 12: Wireframes and illustrations (SSC/N8426: Develop detailed wireframes to illustrate flow, interactions and interface elements) Module 13: Visual design (SSC/N8427: Develop a visual design/screen for the proposed wireframe) Module 14: Design of UI elements (SSC/N8428: Design the assets such as imagery, graphics and animations for the user interface) Module 15: Usability tests (SSC/N8429: Conduct usability tests on the planned user experience designs at different stages) Module 16: User Accessibility for Inclusive Experience Design (SSC/N8455: Design Accessible and Inclusive User Experiences)	24 T & 48 P
Unit 4	DGT/VSQ/N0101: Employability Skills Module 17: Introduction to Employability Skills Module 18: Constitutional values –Citizenship Module 19: Becoming a Professional in the 21st Century Module 20: Basic English Skills Module 21: Communication Skills	10 T & 16 P

	Module 22: Diversity & Inclusion Module 23: Financial and Legal Literacy Module 24: Essential Digital Skills Module 25: Entrepreneurship Module 26: Customer Service Module 27: Getting ready for apprenticeship & Jobs	
Pedagogy:	1. Normal Teaching-learning methodologies, methodologies suggested in the Qualification Pack, Model Curriculum and Courseware 2. Use of the following tools for Practical sessions a. Design and Prototyping: Penpot, Inkscape b. Wireframing Tools: Pencil Project c. User Testing Tools: Lookback.io d. Figma (Optional) 3. One or more of the following learning methodologies. a. Video or Animation b. Collaborative, peer, flipped learning etc. c. Problem-based learning which foster student's analytical skills and develop design thinking skills. d. MOOCs can be recommended for more in depth and advanced understanding of the topics.	
References/ Readings:	Main Reading: 1. IT-ITeS Sector Skills Council NASSCOM. (2024, June). <i>Participants Handbook – User Experience Designer (Version 3.0)</i> Recommended Supplemental Reading: 1. Don Norman. (November 2013). <i>The Design of Everyday Things</i>. Basic Books. 2. Joel Marsh (2022). <i>UX for Beginners</i>. O'Reilly. 3. Wilbert O. Galitz (2007). <i>The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques (Third Edition)</i>. Wiley Publishing. 4. Jesse James Garrett (2011). <i>The Elements of User Experience: User Centered Design for the Web and Beyond (Second Edition)</i>. Pearson Education. 5. Russ Unger and Carolyn Chandler (2012). <i>A Project Guide to UX Design: For user experience designers in the field or in the making (Second edition)</i>. New Riders Publishing USA. 6. Jeff Johnson. (2014). <i>Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines</i>. Morgan Kaufmann.	

**Course
Outcomes:**

At the end of this course, participants will be able to:

1. Describe the nature of work within the IT-ITeS sector, outlining its various sub-sectors and the different occupations found within the Future Skills sub-sector.
 2. Discuss the different web technologies, their historical development, use cases, and business applications.
 3. Explore various mobile technologies, detailing their use cases, and demonstrate the application of different tools, frameworks, platforms, libraries, and software packages to test both hardware and software systems.
 4. Outline the user experience design process by detailing the steps in user experience research, empathy mapping, and understanding user behavior.
 5. Explain how to create a style guide, a design system, and user interface specifications for both developers and designers.
 6. Identify business requirements to extract the user experience needs of customers.
 7. Analyze various trends related to products, customers, and UI design using established methodologies.
 8. Define user personas and behavioral patterns by gathering user behavior data, onboarding unbiased study groups, categorizing users into segments, and developing specific user personas.
 9. Organize information taxonomy and establish a hierarchy of connections.
 10. Create information architecture, including a site map and navigation system.
 11. Develop a user experience map to outline the user journey and flow using standard practices and templates.
 12. Generate wireframes and product illustrations utilizing standard tools and methodologies.
 13. Apply visual design principles to create the product's visual elements and various interaction modes across devices while ensuring a consistent user experience.
 14. Design user interface elements for the product using standard tools and templates.
 15. Conduct usability testing of the software product by creating test cases, defining key performance indicators (KPIs), and monitoring the software's performance and behavior.
 16. Support the team in enhancing performance and achieving goals through effective resource planning, ongoing monitoring, and clear communication.
- Foster new workplace alliances by collaborating with stakeholders and maintaining existing relationships to ensure stakeholder satisfaction.

Qualification Pack, Model Curriculum & Courseware web links	Qualification Pack: https://mozilla.github.io/pdf.js/web/viewer.html?file=https%3A%2F%2Ffs3.ap-south-1.amazonaws.com%2Finsdms2.0%2FStandard%2520Management%2FUser%2520Experience%2520Designer_Version%25203.0%2FQualification%2520Pack%2FSSC_Q8404_v3.0.pdf Model Curriculum: https://mozilla.github.io/pdf.js/web/viewer.html?file=https%3A%2F%2Ffs3.ap-south-1.amazonaws.com%2Finsdms2.0%2FStandard%2520Management%2FUser%2520Experience%2520Designer_Version%25203.0%2FModel%2520Curriculum%2FMC_English_SSCQ8404_User%2520Experience%2520Designer_v3.0.pdf Courseware: Participants Handbook https://mozilla.github.io/pdf.js/web/viewer.html?file=https%3A%2F%2Ffs3.ap-south-1.amazonaws.com%2Finsdms2.0%2FStandard%2BManagement%2FUser%2BExperience%2BDesigner_Version%2B3.0%2FStandards%2BPH%2FPH_English_User%2BExperience%2BDesigner_SSC_Q8404_Version%2B3.0.pdf Facilitator Guide https://mozilla.github.io/pdf.js/web/viewer.html?file=https%3A%2F%2Ffs3.ap-south-1.amazonaws.com%2Finsdms2.0%2FStandard%2BManagement%2FUser%2BExperience%2BDesigner_Version%2B3.0%2FStandards%2BFG%2FFG_English_User%2BExperience%2BDesigner_SSC_Q8404_Version%2B3.0.pdf
---	---

Project (on/off Campus)

Minimum Duration: 90 Hours i.e. equivalent to 1.5 Months to 2 Months.

It is required by a student to carry-out and complete

- On/ Off-Campus Project in the relevant skill area (User Experience Design).
- The project can be done individually or in a group of maximum 5 students.
- There shall be an internal project guide and external, if required.
- After completion of the project, all the artifacts along with the project report in relevant format to be submitted in the college for evaluation and feedback.

[\(Back to Index\)](#)

SEMSTER IV

Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-203
Title of the Course	: Software Testing & Quality Assurance
Number of Credits	: 3 (2T + 1P)
Effective from AY	: 2026-27

Prerequisites for the Course:	None	
Course Objectives:	1. To understand software engineering concepts, software testing principles and Agile Methodologies. 2. To apply test case design methods and software metrics to measure coverage within the SDLC. 3. To analyze defects and software quality using appropriate testing metrics. 4. To evaluate software quality models and Software Quality Assurance processes, for process improvement.	
Unit	Content	No of Hours
I	Introduction : Software Engineering - Concept, Characteristics, Software Life Cycle Model Software Testing : Fundamentals of Testing - Purpose, Principles and Testing Process, Verification vs Validation, Error, Fault and Failure. Testing Types and Approaches - White box testing, Black box Testing, Unit, Integration, System, Acceptance, Regression, Mutation Testing, Non-Functional Testing. Test Case Design : Unit Test Case Design Test Metrics: Introduction, Importance and Types. Introduction to Agile and Scrum Agile Methodology Agile Software Development, Traditional Model vs. Agile Model, Classification of Agile Methods Agile Processes: Work Products, Roles, and Practices, SCRUM, SCRUM Meetings, SCRUM Artifacts, SCRUM Events, Scrum Ceremonies, Crystal, Feature Driven Development, Adaptive	23

	Software Development, Kanban, Extreme Programming, Lean Production	
II	Quality Assurance : Introduction to Software Quality: Quality Concepts, Quality Factors, Software Quality Models (McCall). Software Quality Assurance Process: Introduction, Quality Planning, Quality Control. Software Quality Activities (QA Activities): Software Reviews, Review Types - Requirements, Design, Code, Test artifact, Formal Technical Reviews (FTR). Quality Assurance Process and Standards: Six Sigma, CMM and ISO. Software Quality Assurance Metrics: Key importance, types.	7
III	Practical Work:	30
Week 1 & 2	Introduction to Stakeholders Requirements and Test Requirement Specifications.	02
Week 3	Creation of Manual Test Cases for a given software project	02
Week 4	Introduction to Jira (or any other software) for Testing and Reporting. SCRUM: Creation of Project, Product backlog, Sprint creation & adding stories to sprint	04
Week 5	Selenium Automation Testing : Installation of IDE/Webdriver	02
Week 6-15	Selenium Testing : Write a script to: • Login a specific web page • Update 10 student records into table into excel file • Select number of students who have scored more than 60 in any subject. • Provide a total number of objects available on the page. • Get list of items in a list/combo box • Count number of checkboxes on the page checked and unchecked count • Automate form submission • Verify file upload & download • Handle pagination	20

Pedagogy:	Apart from the normal teach-learning methodology, the following are suggested. 1. Video /animation to explain various concepts 2. Collaborative, peer, flipped learning etc. 3. Adopt problem-based learning which fosters student's analytical skills and develops design thinking skills.
References/ Readings:	Recommended Text Books: 1. Pressman, R. S. (2009). <i>Software engineering: A practitioner's approach</i> (7th ed.). Tata McGraw-Hill. 2. Desikan, S., & Ramesh, G. (2006). <i>Software Testing: Principles and Practices</i>. Pearson Education. Recommended References: 1. Sommerville, I. (2007). <i>Software engineering</i>. Pearson Education Asia. 2. Van Vliet, H. (2008). <i>Software engineering: Principles and practices</i>. Wiley. 3. Jorgensen, P. C. (2014). <i>Software testing: A craftsman's approach</i> (4th ed.). CRC Press: Boca 4. Copeland, L. (2004). <i>A practitioner's guide to software test design</i>. Artech House. 5. Myers, G. J. (2011). <i>The art of software testing</i> (3rd ed.). John Wiley & Sons.
Course Outcomes:	On completion of the course, students will be able to: 1. Understand and explain the basic fundamentals of software testing and Agile Methodologies. 2. Apply and Analyze Software Quality Assurance concepts, quality models , QA activities and review techniques in software development processes. 3. Design and develop basic test cases for effective software testing using tools.

[\(Back to Index\)](#)

Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-204
Title of the Course	: Python Programming
Number of Credits	: 3 (2T + 1P)
Effective from AY	: 2026-27

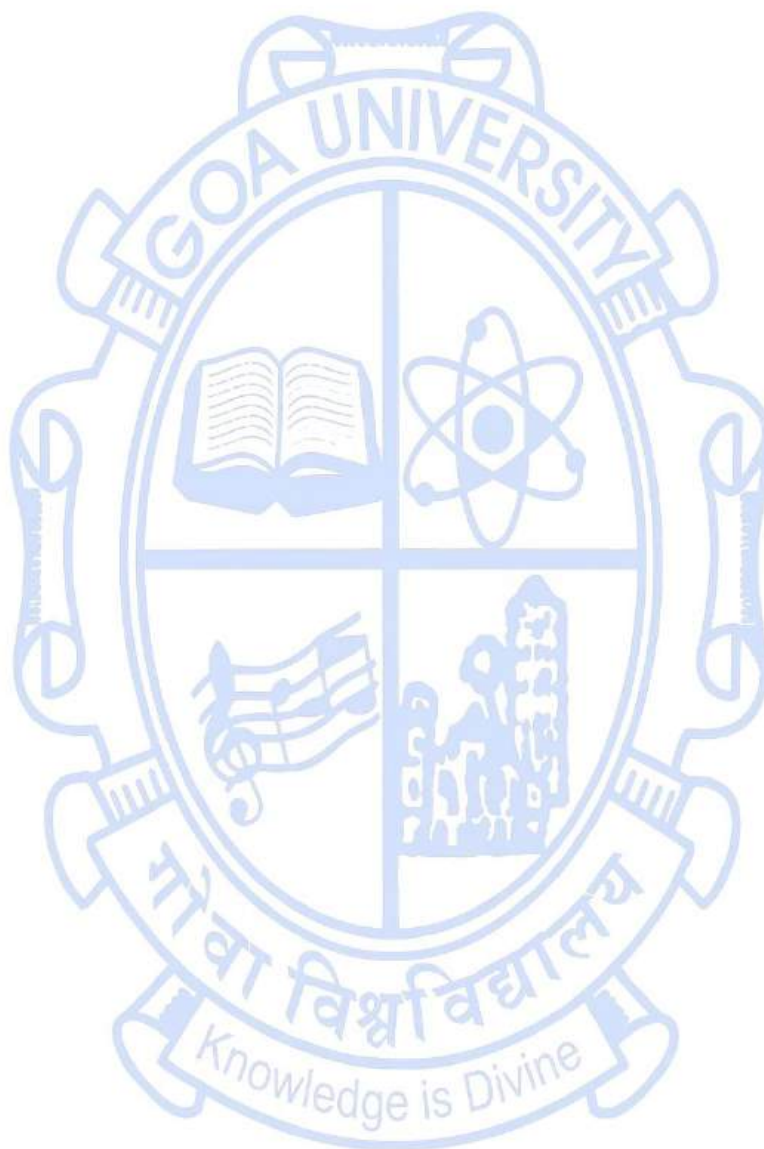
Pre-requisites for the Course:	Object Oriented Concepts	
Course Objectives:	1. Understand the fundamental concepts of Python programming, including data types, variables, operators, control structures, and functions. 2. Apply Python programming constructs such as strings, lists, tuples, and dictionaries to develop simple programs. 3. Implement file handling, exception handling, and modular programming techniques in Python applications. 4. Analyze problems and develop Python programs using object-oriented concepts and libraries such as NumPy, Pandas, and Matplotlib.	
Units		No of hours
I	INTRODUCTION TO PYTHON Python interpreter and interactive mode; values and types: int, float, Boolean, string, and list; variables, expressions, statements, assignment, precedence of operators, comments; Conditionals: Boolean values and operators, conditional (if), alternative (if-else), chained conditional (if-elif-else). Iteration statements: while, for, break, continue, pass. Functions: function definition and use, flow of execution, parameters and arguments, local and global scope, fruitful functions, recursion.	08
II	STRINGS, LIST, TUPLE, DICTIONARY Strings: immutability, string slicing, string functions and methods. Lists: list operations, list slicing, list methods, list loop, mutability, aliasing, cloning lists, list parameters, advanced list processing - list comprehension; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods.	10
III	FILES, EXCEPTION HANDLING, OBJECT ORIENTED CONCEPTS and PYTHON LIBRARIES Files and Exception: text files, reading and writing files,	15

	command line arguments, errors and exceptions, handling exceptions. Object Oriented concepts- classes, objects, abstract data types, polymorphism, encapsulation, mutator and accessor methods, static methods, adding methods dynamically, inheritance and introduction to composition, built-in functions for classes. Exploring Python Libraries - NumPy , Pandas and Matplotlib.	
IV	Practical Work	30
Week 1 &2	Introduction to Python & Data Types • Installation and working with Python (Interactive & Script mode) • Variables, data types: int, float, bool, string • Type casting • Operators and precedence • Basic input/output	04
Week 3 & 4	Conditionals and Iterative Statements • if, if-else, if-elif-else • Match case • Nested conditionals • for loop, while loop • break, continue, pass	04
Week 5 & 6	Functions and Strings • Function definition and calling • Parameters and return values • Local and global variables • Recursion • String indexing, slicing, methods	04
Week 7 & 8	Lists and Advanced List Processing • List operations and slicing • List methods • Mutability, aliasing, cloning • List comprehension • Nested lists	04
Week 9 & 8	Tuples, Dictionaries and File Handling • Tuple operations • Tuple as return value • Dictionary operations and methods • Reading and writing text files • File modes	04

Week 9 & 10	Exception Handling and Modules • Types of errors and exceptions • try, except, finally • Multiple except blocks • Raising exceptions • Command line arguments • Creating and importing modules	04
Week 11 & 12	Object-Oriented Concepts • Class and object creation • Constructor (init) • Instance and static methods • Inheritance and polymorphism	04
Week 15	Python Libraries • Introduction to NumPy, Pandas, Matplotlib • Basic data visualization	02
Pedagogy:	Suggested strategies for use to accelerate the attainment of the various course outcomes are: 1. Video/animation to explain various concepts. 2. Collaborative, peer, flipped learning 3. Adopt problem-based learning which fosters student's analytical skills and develops design thinking skills.	
References/ Readings:	Text Books 1. Downey, A. B. (2016). <i>Think Python: How to think like a computer scientist</i> (Updated for Python 3). Shroff/O'Reilly Publishers. 2. van Rossum, G., & Drake, F. L. Jr. (2011). <i>An introduction to Python: Revised and updated for Python 3.2</i>. Network Theory Ltd. Reference Books 1. John V Guttag, Introduction to Computation and Programming Using Python“, Revised and expanded Edition, MIT Press , 2013 2. Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016. 3. Timothy A. Budd, Exploring Python, Mc-Graw Hill Education (India) Private Ltd., 2015. 4. Kenneth A. Lambert, Fundamentals of Python: First Programs, CENGAGE Learning, 2012. 5. Charles Dierbach, Introduction to Computer Science using Python: A Computational Problem Solving Focus, Wiley India Edition, 2013. 6. Paul Gries, Jennifer Campbell and Jason Montojo, Practical Programming: An Introduction to Computer Science using Python 3, Second edition, Pragmatic Programmers, LLC, 2013.	

Course Outcomes:	At the end of this course, the students will be able to: 1. Understand basic Python concepts such as data types, variables, operators, control structures, and functions. 2. Apply Python constructs like strings, lists, tuples, and dictionaries to write programs. 3. Implement file handling, exception handling, and modules in Python. 4. Analyze and develop programs using OOP concepts and Python libraries like NumPy, Pandas, and Matplotlib.
-------------------------	---

[\(Back to Index\)](#)



Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-205
Title of the Course	: Statistics
Number of Credits	: 2T
Effective from AY	: 2026-27

Pre-requisites for the Course:	None	
Course Objectives:	1. To define, describe, and explain fundamental concepts of statistics, types of data, and methods of data organization and visualization. 2. To apply descriptive statistical techniques and probability concepts for analysing structured and unstructured data. 3. To analyze data using measures of dispersion, skewness, sampling techniques, and statistical relationships such as correlation and regression. 4. To evaluate statistical outcomes obtained from hypothesis testing and interpret results for informed decision-making. 5. To design and develop basic statistical models and data representations using appropriate tools and techniques.	
Units		No of hours
I	DATA UNDERSTANDING AND DESCRIPTIVE STATISTICS Role of statistics in computing and data science, Types of data. (qualitative and quantitative), Structured vs Unstructured data, Class intervals, Frequency Tables, Tabular representation of data, Bar diagrams and their types; Pie charts; Frequency polygon; Histogram; Ogives. Positional Measures: Percentiles, quartiles, quartile deviation, Measures of variability: coefficient of variation. Shape of distributions: skewness and interpretation. Introduction to Probability Random experiments and sample space, Events and basic probability rules, Conditional probability, Bayes' theorem and application.	15
II	SAMPLING AND HYPOTHESIS TESTING Sampling Techniques: Simple Random, Stratified, Cluster Sampling Distribution and Confidence Intervals Hypothesis Testing: t-test, Chi-square Test Parametric and Non-parametric Tests (Conceptual) CORRELATION AND REGRESSION ANALYSIS Karl Pearson Correlation Coefficient and Interpretation	15

	Simple Linear Regression and Regression Equation	
Pedagogy:	Conceptual lectures, demonstration of data organization and visualization techniques, numerical problem solving, case-based discussions, interpretation of statistical results, and demonstration using spreadsheet or statistical tools.	
References/ Readings:	Text Books 1. S. C. Gupta: <i>Fundamentals of Statistics</i>, 7th Edition, Himalaya Publishing House, 2018. 2. S. C. Gupta, and V. K. Kapoor: <i>Fundamentals of Mathematical Statistics</i>, 12th Edition, S. Chand and Sons, Delhi, 2020. Reference Books 1. A. M. Goon, M. K. Gupta, and B. Dasgupta: <i>Fundamentals of Statistics, Vol. I</i>, 8th Edition, The World Press, Kolkata, 2016. 2. S. P. Gupta: <i>Statistical Methods</i>, S. Chand & Sons, 2017. 3. S. Bernstein, and R. Bernstein: <i>Schaum's Outlines: Elements of Statistics I – Descriptive Statistics and Probability</i>, McGraw Hill, 2020.	
Course Outcomes:	At the end of this course, the students will be able to: 1. Define and identify types of data, statistical concepts, probability terms, and sampling methods. 2. Explain and illustrate data organization techniques, graphical representations, and descriptive statistical measures. 3. Apply statistical methods to compute quartiles, percentiles, coefficient of variation, and basic probability. 4. Analyze datasets using sampling techniques, hypothesis testing (t-test, chi-square), correlation, and regression. 5. Evaluate and interpret statistical results to support data-driven decision-making.	

[\(Back to Index\)](#)

Name of the Programme	: Bachelor of Vocation in Software Technologies
Course Code	: STG-222 SSC/Q8403: Application Developer – Web and Mobile (Version 3.0)
Title of the Course	SSC/Q8403: Application Developer – Web and Mobile (Version 3.0)
Number of Credits	: 12 (5 Theory, 4 Practical, 3 Project)
Effective from AY	: 2026-27

Pre-requisites for the Course:	Basic knowledge of programming concepts, computer networks, databases, and familiarity with web or mobile application development fundamentals.	
Course Objectives:	1. SSC/Q8403: Application Developer - Web & Mobile 2. SSC/N8417: Implement DevSecOps practices for secure applicationDevelopment 3. SSC/N8451: Implement Continuous Integration and ContinuousDeployment (CI/CD) practices 4. SSC/N8125: Develop tests or simulations for end-to-end QA of systems 5. SSC/N8418: Fix application bugs and improve application performance 6. SSC/N8323: Monitor and manage cloud applications and the deployed systems 7. SSC/N8452: Optimize Revenue, Security, and Accessibility in Web andMobile Products 8. DGT/VSQ/N0102: Employability Skills 9. SSC/N8414: Develop a user-friendly web app aligned with functional, nonfunctional, and UX requirements 10. SSC/N8415: Develop native/cross-platform/hybrid mobile application forthe target platforms 11. SSC/N8416: Develop reliable, scalable and secure back-end aligned to the application architecture	
UNITS	Content:	No. of Hours 285
Unit 1	Module 1 : Introduction to the IT-ITeS/BPM Industries and the Job Role of an Application Developer – Web & Mobile - Introduction to IT-ITeS/BPM Industries - Job Role: Application Developer – Web & Mobile - Basics of Web Technology - Basics of Mobile Technology - Future Trends & Career Growth	16T & 18P

	(SSC/Q8403: Application Developer - Web & Mobile) Module 2: Securing Application Development with DevSecOps - Organizational Security Policies & DevSec Ops Foundations - Product Roadmaps, Requirements & Competitive Analysis - Secure Version Control & Repository Management - Security Tools & Compliance - Generative AI in Product Management (SSC/N8417: Implement DevSecOps practices for secure application Development)	
Unit 2	Module 3: Implementing Continuous Integration and Continuous Deployment (CI/CD) - Product Development Life Cycles & Roadmapping - Product Metrics & Monitoring - CI/CD Concepts & Tools - Build Automation & Deployment Best Practices (SSC/N8451: Implement Continuous Integration and Continuous Deployment (CI/CD) practices) Module 4: End-to-End Testing and Simulations for System Quality Assurance - Requirements & Test Strategy Foundations - Test Design Principles & Types of Testing - Simulation Tools & Automation Feasibility - Test Execution, Documentation & Feedback Loop - Performance & Load Testing Concepts (SSC/N8125: Develop tests or simulations for end-to-end QA of systems) Module 5: Application Bug Fixing and Performance Enhancement - Requirements Analysis & Testing Context - Logging, Log Analysis & Organizational Policies - Bug Identification, Root Cause Analysis & Documentation - Testing Tools, Automation & Debugging Techniques - Feedback, Communication & Performance Enhancement (SSC/N8418: Fix application bugs and improve application performance)	25 T & 38 P

Unit 3	Module 6: Cloud Application Monitoring and System Management - Organizational Monitoring Policies & Governance - Application Performance Metrics & Resource Monitoring - Monitoring Tools & Technologies - Reporting, Analysis & Decision-Making (SSC/N8323: Monitor and manage cloud applications and the deployed systems) Module 7: Optimizing Revenue, Security, and Accessibility in Web and Mobile Products - Revenue Optimization & Monetization Models - Product Lifecycle, Customer Insights & Prioritization - Security, Privacy & Compliance - Accessibility, Analytics & Cross-Functional Coordination (SSC/N8452: Optimize Revenue, Security, and Accessibility in Web and Mobile Products) Module 9: User-Friendly Web Application Development - Project Scope & Development Lifecycle - Application Architecture & Microservices - CI/CD, Containers & Automation - Security, Compliance & Data Protection - Data Management & Backup Strategies - Accessibility & Agile Methodologies - Encryption & Secure Development (SSC/N8414: Develop a user-friendly web app aligned with functional, nonfunctional, and UX requirements) Module 10: Developing Native, Cross- Platform, and Hybrid Mobile Applications - Requirements Mapping & Development Lifecycle - Native vs Hybrid vs Cross-Platform Development - UX/UI Design Principles - Security, Encryption & Compliance - Automation, CI/CD & App Store Publishing - Emerging Technologies & Scaling - Independent App Development Strategies (SSC/N8415: Develop native/cross-platform/hybrid mobile application for the target platforms)	24 T & 48 P
Unit 4	Module 8: Employability Skills (DGT/VSQ/N0102: Employability Skills) - Introduction to Employability Skills	10 T & 16 P

	- Constitutional values - Citizenship - Becoming a Professional in the 21st Century - Basic English Skills - Career Development & Goal Setting - Communication Skills - Diversity & Inclusion - Financial and Legal Literacy - Essential Digital Skills - Entrepreneurship - Customer Service - Getting Ready for apprenticeship & Jobs Module 10: Developing Native, Cross-Platform, and Hybrid Mobile Applications Module 11: Developing a Reliable, Scalable, and Secure Back-End Architecture	
Pedagogy:	1. Normal Teaching-learning methodologies, methodologies suggested in the Qualification Pack, Model Curriculum and Courseware 2. One or more of the following learning methodologies. a. Video or Animation b. Collaborative, peer, flipped learning etc. c. Problem-based learning which foster student's analytical skills and develop design thinking skills. d. MOOCs can be recommended for more in depth and advanced understanding of the topics. 3. Live demonstrations of tools and workflows commonly used in industry. Examples include: • Git version control workflows • CI/CD pipeline configuration • containerized deployment using Docker • application monitoring dashboards • Git and GitHub for version control • CI/CD tools such as Jenkins or GitLab CI container technologies like Docker	
References/ Readings:	Main Reading: IT-ITeS Sector Skills Council NASSCOM. (2024, June). <i>Participants Handbook – IT-ITeS Sector Skills Council NASSCOM. (2024, June). Participants Handbook – Application Developer Web & Mobile (Version 3.0)</i> Recommended Supplemental Reading:	

	Continuous Delivery – Jez Humble and David Farley, <i>Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation</i>, Addison-Wesley. The DevOps Handbook – Gene Kim, Jez Humble, Patrick Debois, and John Willis, <i>The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations</i>, IT Revolution Press.
Course Outcomes:	At the end of this course. participants will be able to: 1. DevSecOps & Software Development Concepts 2. Requirement Analysis & System Design 3. Web & Mobile Application Development 4. Testing & Quality Assurance 5. Debugging & Performance Optimization 6. Cloud Monitoring & System Management 7. Revenue Optimization, Security & Accessibility
Qualification Pack, Model Curriculum & Courseware web links	Qualification Pack: Application Developer - Web & Mobile Model Curriculum: Courseware: Participants Handbook https://s3.ap-south-1.amazonaws.com/insdms2.0/Standard%20Management/Application%20Developer%20-%20Web%20%26%20Mobile_Version%203.0/Standards%20PH/PH_English_Application%20Developer-Web%20%26%20Mobile_SSC_Q8403_V3.0.pdf#pdfjs.action=download Facilitator Guide https://s3.ap-south-1.amazonaws.com/insdms2.0/Standard%20Management/Application%20Developer%20-%20Web%20%26%20Mobile_Version%203.0/Standards%20FG/FG_English_Application%20Developer%20-%20Web%20%26%20Mobile_SSC_8403_V3.0.pdf#pdfjs.action=download

Project (on/off Campus)

Minimum Duration: 90 Hours i.e. equivalent to 1.5 Months to 2 Months.

It is required by a student to carry-out and complete

- On/ Off-Campus Project in the relevant skill area (User Experience Design).
- Project can be done individually or in a group of maximum 5 students.
- There shall be an internal project guide and external, if required.
- After completion of the project, all the artifacts along with the project report in relevant format to be submitted in the college for evaluation and feedback.

(SSC/N8416: Develop reliable, scalable and secure back-end aligned to the application architecture)

Name of the Programme	: B.Voc. (Software Technologies)
Course Code	: STG-261
Title of the Course	: Web Application Development
Number of Credits	: 4 (2 Theory + 2 Practical)
Effective from AY	: 2026-27

Pre-requisites for the Course:	Basic Programming, Object Oriented Concepts and DBMS Courses	
Course Objectives:	1. To understand the fundamentals of client-side and server-side technologies. 2. To understand dynamic and interactive web experiences using JavaScript and client-side frameworks. 3. To design web applications using server-side technologies and databases.	
Units	Content:	No. of hours (30T + 60P)
I	Client-Side Scripting Framework • Introduction to client-side scripting language, Variables types – var, let and const, data types, Operators. • Conditional statements, Loops, Functions (Named functions, anonymous functions, arrow functions) • Template literals, Destructuring, Spread/Rest operators • Promises/async-await • DOM (Document Object Model) • Event Handling – Attaching events to HTML elements • Introduction to CSS frameworks, Integrating Bootstrap into web application, Understanding Bootstrap grid system. • Bootstrap – Containers, Carousal, navbar, tables, forms, images, typography, color	15
	Server-Side framework and Database Connectivity • Introduction to server-side scripting language, Input/Output statements, Decision Statements, Looping Statements. • Database Connectivity, CRUD (Create, Update, Read and Delete). • Introduction to server-side frameworks, Downloading and installing server-side framework, Directory Structure, modules, libraries.	
II		15

	• APIs, configuring database connections.	
III	Practical Activities - To be carried out along in sync with the concepts mentioned in Unit I & II respectively. • Set up a development environment using VS Code and Node.js, and develop a strong foundation in JavaScript including variables, data types, naming conventions, operators, conditional statements, and looping constructs. • Write and use different types of functions in JavaScript, including named, anonymous, and arrow functions to develop modular, reusable, and structured code. • Understand the DOM structure, select HTML elements, dynamically create and modify content, handle events, and implement client-side form validation using JavaScript. • Integrate Bootstrap into web applications and build responsive layouts using the grid system, containers, rows, columns, and responsive breakpoints. • Design interactive and visually appealing web pages using Bootstrap components such as Navbar, Carousel, Cards, Buttons, Badges, Alerts, Tables, and Forms, along with utility classes for styling and layout control. • Provide practical knowledge of server-side scripting using PHP, including variables, operators, control structures, functions, and to establish secure database connectivity with MySQL using mysqli. • Develop the ability to perform CRUD operations using PHP and MySQL, build and process HTML forms, create simple API endpoints returning JSON data, and integrate JavaScript, Bootstrap, PHP, and MySQL into a complete full-stack web application.	40
IV	Case Studies- 1. Student Registration and Management System Background 2. Online Product Catalog with Admin Panel Mini project- Develop a mini project by applying the insights gained from the case studies to a real-world scenario. Optional -Prepare for industry-recognized certifications by taking practice exams, completing online courses, and participating in certification programs offered by platforms like skillshare, freeCodeCamp, W3schools, Microsoft(Azure/Web Dev), Meta (Front-End Developer) or Oracle (Database). It will enhance the credentials and	20

	increase the employability in the web development and software industry.	
Pedagogy	Suggested strategies for use to accelerate the attainment of the various course outcomes. 1. A plan is to be developed by the student/s in consultation with the teacher incharge and to be approved. 2. One or more methods mentioned below may be used for learning purposes. a. Intensive training / teaching b. Online or offline training (approved by the college or instructor) c. Approved MOOCS Courses d. Workshops - on-campus or off-campus e. Self-learning means & methods f. Enquiry-based learning 3. A work diary to be maintained where all the learning & work carried out to maintained and certified by the teacher incharges. 4. All deliverable & artifacts to be submitted in the college for evaluation and assessments.	
References:	Main Reading: 1. Haverbeke, M. (2018). Eloquent JavaScript: A Modern Introduction to Programming (3rd ed.). No Starch Press. 2. Welling, L., & Thomson, L. (2016). PHP and MySQL Web Development (5th ed.). Pearson Education. 3. Robbins, J. Learning Web Design: A Beginners Guide to HTML, CSS, Javascript and Web graphics. (5th ed.). O'Reilly. Additional Reading: 1. Fielding, J. (2014). Beginning Responsive Web Design with HTML5 and CSS3. Apress. 2. Stauffer, M. (2023). Laravel: Up & Running: A Framework for Building Modern PHP Apps (3rd ed.). O'Reilly. 3. Sullivan, B., & Lui, V. (2012). Web Application Security, A Beginner's Guide. McGraw-Hill Education. 4. Deitel, P. (2018). Internet and World Wide Web-How to Program (5th ed.). Pearson Education.	
Course Outcomes	On completion of the course, students will be able to: 1. Understand and utilize JavaScript for dynamic web behaviors, including DOM manipulation and event handling. 2. Apply a client-side framework for responsive, mobile-first web design components, and grid system to deliver visually appealing and user-friendly web experiences across various devices and screen sizes 3. Design dynamic and interactive web applications to process user requests, interact with databases, manage server-side logic, and generate dynamic content. 4. Build and deploy a complete full stack web application integrating client side , server-side and database technologies.	

[\(Back to Index\)](#)