

<b>Programme</b>	BS Computational Statistics and Data Analytics	<b>Course Code</b>	<b>CSTA-204</b>	<b>Credit Hours</b>	3
<b>Course Title</b>	Introduction to Programming				
<b>Course Introduction</b>					
This course offers a foundational understanding of programming concepts and principles. Students learn fundamental coding constructs, problem-solving techniques, and algorithmic thinking using a high-level programming language. Through hands-on exercises and projects, they acquire the skills necessary to design, implement, and debug simple programs, laying the groundwork for further exploration in the field of computer science.					
<b>Learning Outcomes</b>					
By the end of this course, students will be able to:					
<ol style="list-style-type: none"> <li>1. Understand fundamental programming concepts.</li> <li>2. Learn the basics of Java programming.</li> <li>3. Develop problem-solving skills through coding.</li> <li>4. Apply Java for creating simple applications.</li> </ol>					
<b>Course Content</b>				<b>Assignments/Readings</b>	
<b>Week 1</b>	<b>Unit – I</b> Overview of Programming Concepts: Introduction to programming paradigms and principles				
	<b>Unit – II</b> Basics of problem-solving techniques and algorithm development				
<b>Week 2</b>	<b>Unit – III</b> Problem-solving and Algorithm Development: Strategies for breaking down problems and designing algorithms				
	<b>Unit – IV</b> Techniques for analyzing and improving algorithm efficiency				
<b>Week 3</b>	<b>Unit – V</b> Basics of Java: Introduction to the Java programming language				
	<b>Unit – VI</b> Understanding variables, data types, and operators in Java				
<b>Week 4</b>	<b>Unit – VII</b> Control Structures: Learning control structures such as if statements, switch statements, and loops				
	<b>Unit – VIII</b> Applying control structures in Java programming				
<b>Week 5</b>	<b>Unit – IX</b> Object-Oriented Programming (OOP): Understanding the principles of OOP: encapsulation,				

	inheritance, and polymorphism	
	<b>Unit – X</b> Implementing OOP concepts in Java programming	
<b>Week 6</b>	<b>Unit – XI</b> Control Structures (Continued): Mastering decision-making and looping constructs in Java	
	<b>Unit – XII</b> Advanced control flow techniques and best practices	
<b>Week 7</b>	<b>Unit – XIII</b> Understanding Classes, Objects, and Methods in Java: Definition and usage of classes, objects, and methods in Java	
	<b>Unit – XIV</b> Implementing classes and methods in Java programs	
<b>Week 8</b>	<b>Unit – XV</b> Reading and Writing Data in Java Applications: Input and output operations in Java: reading from and writing to files, streams, and console	
	<b>Unit – XVI</b> Handling data serialization and deserialization in Java	
<b>Week 9</b>	<b>Unit – XVII</b> Control Structures (Continued): Advanced decision-making and looping techniques	
	<b>Unit – XVIII</b> Nested loops, labeled loops, and loop control statements	
<b>Week 10</b>	<b>Unit – XIX</b> Exception Handling: Introduction to exceptions and error handling in Java Writing robust code using try-catch blocks and exception propagation	
	<b>Unit – XX</b> Functions and Methods: Understanding functions and methods in Java	
<b>Week 11</b>	<b>Unit – XXI</b> Creating and using functions and methods for code modularity and reusability	
	<b>Unit – XXII</b> Advanced Methods and Functionality: Method overloading, overriding, and visibility modifiers	
<b>Week 12</b>	<b>Unit – XXIII</b> Working with Arrays and Collections in Java: Introduction to arrays and collections in Java	
	<b>Unit – XXIV</b> Using arrays and collections to store and manipulate data efficiently	
<b>Week 13</b>	<b>Unit – XXV</b> Exception Handling (Continued): Advanced	

	exception handling techniques: throwing exceptions, custom exceptions	
	<b>Unit – XXVI</b> Best practices for handling exceptions in Java applications	
<b>Week 14</b>	<b>Unit –XXVII</b> Dealing with Runtime Errors and Exceptions: Debugging techniques for identifying and fixing runtime errors Handling common runtime exceptions in Java programs	
	<b>Unit – XXVIII</b> Introduction to Graphical User Interfaces (GUIs) in Java:Basics of GUI design and development using Java Swing or JavaFX Creating simple GUI applications in Java	
<b>Week 15</b>	<b>Unit – XXIX</b> File Handling: Reading and writing files in Java applications Understanding file I/O operations and file manipulation in Java	
	<b>Unit – XXX</b> Introduction to Data Structures: Overview of fundamental data structures: arrays, linked lists, stacks, and queues Implementation and usage of basic data structures in Java	
<b>Week 16</b>	<b>Unit – XXXI</b> Basics of Data Structures: Understanding more complex data structures like trees, graphs, and hash tables Implementation and manipulation of data structures in Java	
	<b>Unit – XXXII</b> Simple Java Projects: Building and executing simple Java applications and projects Applying learned concepts and techniques to solve real-world problems Debugging and Troubleshooting: Strategies for identifying and fixing code issues in Java programs Using debugging tools and techniques for troubleshooting Java applications	
<b>Textbooks and Reading Material</b>		

**Textbook:**

1. Schildt, H. (2007). Java: the complete reference.
2. Sierra, K., & Bates, B. (2005). Head First Java: A Brain-Friendly Guide. "O' Reilly Media, Inc."

**Suggested Readings:**

1. Liang, Y. D. (2009). Introduction To Java Programming, Comprehensive Version, 7/E. Pearson Education India.
2. Lassooff, M. (2017). Java Programming for Beginners: Learn the Fundamentals of Programming with Java. Packt Publishing.

**Teaching Learning Strategies**

Class Lecture method, which includes seminars, discussions, assignments and projects. (Audio-visual tools are used where necessary)

**Assignments: Types and Number with Calendar**

According to the choice of respective teacher.

**Assessment**

<b>Sr. No.</b>	<b>Elements</b>	<b>Weightage</b>	<b>Details</b>
1.	Midterm Assessment	35%	It takes place at the mid-point of the semester.
2.	Formative Assessment	25%	It is continuous assessment. It includes: Classroom participation, attendance, assignments, and presentations, homework, attitude and behavior, hands-on-activities, short tests, quizzes etc.
3.	Final Assessment	40%	It takes place at the end of the semester. It is mostly in the form of a test, but owing to the nature of the course the teacher may assess their students based on term paper, research proposal development, field work and report writing etc.