

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - CYBER SECURITY

COURSE MODULE OF THE SUBJECT TAUGHT FOR THE SESSION 2026-27 (ODD SEM)

Faculty Name: Dr. Shwetha G K			Academic Year: 2026 - 2027				
Department: Computer Science Engineering -Cybersecurity							
Course Code	Course Title	Core / Elective	Prerequisite	Contact Hours			Total Hrs/
				L	T	P	Sessions
1BCS302	OBJECT ORIENTED PROGRAMMING WITH JAVA	IPCC	Computer Fundamentals	3	0	2	42+28=70
Topics Covered as per Syllabus							
Module-1							
An Overview of Java: Object-Oriented Programming (Two Paradigms, Abstraction, The Three OOP Principles), Using Blocks of Code, Lexical Issues (Whitespace, Identifiers, Literals, Comments, Separators, The Java Keywords). Data Types, Variables, and Arrays: The Primitive Types (Integers, Floating-Point Types, Characters, Booleans), Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays. Operators: Arithmetic Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses. Control Statements: Java’s Selection Statements (if, The Traditional switch), Iteration Statements (while, do-while, for, The For-Each Version of the for Loop, Local Variable Type Inference in a for Loop,Nested Loops), Jump Statements (Using break, Using continue, return). Textbook: Chapters 2, 3, 4, 5							
Module-2							
Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, the 'this' Keyword, and Garbage Collection. Methods and Classes: Overloading Methods, Using Objects as Parameters, Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Introducing Nested and Inner Classes. Textbook: Chapters 6, 7							
Module-3							
Inheritance: Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using `final` with Inheritance, Local Variable Type Inference and Inheritance, The Object Class. Packages: Packages, Packages and Member Access, Importing Packages. Interfaces: Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods. Textbook: Chapters 8, 9							
Module-4							
Exceptions: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java’s Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions. Multithreaded Programming: The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities,Obtaining a Thread’s State. Textbook: Chapters 10, 11							

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - CYBER SECURITY

Module-5

Enumerations, Type Wrappers and Autoboxing: Enumerations (Enumeration Fundamentals, The values() and valueOf() Methods), Type Wrappers (Character, Boolean, The Numeric Type Wrappers), Autoboxing (Autoboxing and Methods, Autoboxing/Unboxing Occurs in Expressions, Autoboxing/Unboxing Boolean and Character Values). Generics: A Simple Generics Example, A Generic Class with Two Type Parameters, The General Form of a Generic Class.

Textbook: Chapters 12, 14

Textbooks:

Herbert Schildt and Danny Coward, Java: The Complete Reference, 13th Edition, McGraw-Hill, 2024.

Reference Books

1. Cay S. Horstmann, Core Java Volume I – Fundamentals, 14th Edition, Oracle Press, 2026.
2. E. Balagurusamy, Programming with Java, 7th Edition, McGraw-Hill Education, 2024.

PRACTICAL COMPONENT OF IPCC

1. Develop a Java program to add TWO matrices of suitable order N.
2. Develop a stack class to hold a maximum of 10 integers with suitable methods. Develop a Java main method to illustrate Stack operations.
3. Create a Java class called Student with the USN, Name, Branch, Phone details as variables within it. Write a Java program to create N Student objects and print the USN, Name, Branch, and Phone of these objects with suitable headings.
4. Develop a Java program to create a class named Shape. Create three sub classes, namely: circle, triangle and square. Each class has two member functions named draw and erase. Demonstrate the concepts of polymorphism by developing appropriate methods, defining member data, and implementing the main program.
5. Develop a Java program to create an interface Resizable with methods resizeWidth(int width) and resizeHeight(int height) that allow an object to be resized. Create a class Rectangle that implements the Resizable interface and implements the resize methods.
6. Develop a Java program to create a package named mypack and import & implement it in a suitable class.
7. Develop a Java program to raise a custom exception (user-defined exception) for DivisionByZero using try, catch, throw and finally.
8. Develop a program to create a class MyThread with a class constructor, call the base class constructor, using super and start the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently.
9. Develop a Java program using an enumeration called Department containing CSE, ISE, ECE, and MECH. Accept a department name as input and display the selected department using valueOf().
10. Develop a Java program to read the marks of three subjects as strings, convert them into wrapper objects, calculate the total and average using autoboxing/unboxing and display the result.
11. **Micro Project:** A team of two to three students shall develop a project and submit a brief report for a specific task assigned by the course instructor based on the concepts covered in the syllabus.

Note: Each experiment (1 to 10) will be evaluated for 30 marks (total 300), and the micro-project for 120 marks. Total marks (420) shall be scaled down to 15. Make use of the most recent version of the IDE tool for program development.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - CYBER SECURITY

Suggested Learning Resources: (Text Book/ Reference Book/ Manuals): Textbooks:

1. Herbert Schildt and Danny Coward, Java: The Complete Reference, 13th Edition, McGraw-Hill, 2024.

Reference books / Manuals:

1. Cay S. Horstmann, Core Java Volume I – Fundamentals, 14th Edition, Oracle Press, 2026.
2. E. Balagurusamy, Programming with Java, 7th Edition, McGraw-Hill Education, 2024

Web links and Video Lectures (e-Resources):

1. Official Java documentation (Oracle): <https://docs.oracle.com/javase/tutorial/>
2. Modern Java learning resources (Dev Java): <https://dev.java/learn/>
3. Java Tutorial: <https://www.w3schools.com/java/>
4. Programming examples and practice (GeeksforGeeks): <https://www.geeksforgeeks.org/java/java/>
5. Infosys Springboard Course on Java Programming Fundamentals: <https://infyspringboard.onwingspan.com/web/en/login>
6. HackerRank platform: <https://www.hackerrank.com/domains/java>

Structured video lectures (NPTEL): (<https://nptel.ac.in/courses/106105191/>)

Teaching-Learning Process (Innovative Delivery Methods):

The following are sample strategies that educators may adopt to enhance the effectiveness of the teaching-learning process and facilitate the achievement of course outcomes.

1. Flipped Classroom.
2. Problem-Based Learning
3. ICT Enabled Teaching
4. Programming Assignments

Tool-Based Learning using Eclipse/IntelliJ IDEA

Course outcomes: The students should be able to:

- Demonstrate proficiency in writing simple programs involving branching and looping structures.
- Construct object-oriented solutions using classes, objects, and methods.
- Develop programs using inheritance, packages, and interfaces.
- Make use of exception handling and multithreading in solving complex problems
- Apply concepts of autoboxing, enumerations, and generics in program development.
- Choose a suitable IDE tool to design, develop, and debug Java programs.

Assessment Details (both CIE and SEE)

Assessment Structure: The assessment for each course is equally divided between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each component carrying 50% weightage (i.e., 50 marks each).

The CIE Theory component will be 25 marks and CIE Practical component will be 25 marks. The CIE Theory component consists of IA tests for 25 marks.

The CIE Practical component for continuous assessments will be for 15 marks through rubrics and for lab tests will be for 10 marks.

- To qualify and become eligible to appear for SEE, in the CIE theory component, a student must score at least 40% of 25 marks, i.e., 10 marks.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - CYBER SECURITY

- To qualify and become eligible to appear for SEE, in the CIE Practical component, a student must secure a minimum of 40% of 25 marks, i.e., 10 marks.
- To pass the SEE, a student must secure a minimum of 35% of 50 marks, i.e., 18 marks.
- A student is deemed to have successfully completed the course if the combined total of CIE and SEE is at least 40 out of 100 marks.

CIE Practical component: The CIE marks awarded in the case of the Practical component shall be based on the continuous evaluation of the laboratory report using a defined set of rubrics. Each experiment report can be evaluated for 30 marks. The summation of all the experiments marks to be scaled down to 15 marks. The laboratory test (duration 03 hours) at the end of the last week of the semester /after completion of all the experiments (whichever is early) shall be conducted for 50 marks and scaled down to 10 marks. For the laboratory test, the student is required to conduct one experiment (excluding the micro project).

Continuous Internal Evaluation (CIE)

- Theory component** The CIE marks of 25 shall be earmarked for two tests. The first test shall be conducted after completing two modules of the syllabus and the second one after completing the rest three modules. Each test shall be conducted for 25 marks. The average of the two tests scaled down to 25 marks shall constitute the test marks for a maximum of 25 marks.
- Laboratory component** Out of 25 marks, 15 marks shall be assigned for assessment as per the rubrics listed in the course syllabus (30 marks scaled down to 15). The remaining 10 marks shall be based on the practical test conducted by two Internal examiners appointed by the HoD. The allotment of marks shall be as per the rubrics defined for the practical test (50 marks scaled down to 10).
 - To qualify and become eligible to appear for the SEE of the IC/IPCC, a student shall secure at least 40 % of 25 marks, i.e., 10 marks, in the CIE theory component, and at least 40 % of 25 marks, i.e., 10 marks in the CIE Practical component.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - CYBER SECURITY

Passing requirement in SEE

For a pass in the Semester End Examination (SEE), a student shall secure a minimum of 35 marks out of 100. For the declaration of a pass grade, the sum of the Continuous Internal Evaluation (CIE) marks (out of 50) marks and the SEE marks scaled down to 50 shall be greater than or equal to 40 marks. If the total (CIE + SEE) is less than 40 marks, the student shall be awarded the grade 'F' (Fail).

Rubrics for CIE of Practical Component:

Criteria/ Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Unacceptable (1)
Fundamental Knowledge: Understanding the problem statement [CO1] [PO1, PO2]	The student has in depth knowledge of the topics related to the problem. Student is able to completely understand the problem definition.	Student has good knowledge of some of the topics related to problem. Student is able to understand the problem definition.	Student is capable of narrating the answer but not capable to show in depth knowledge and the problem definition.	Student has not understood the concepts partially. Student is able to partially understand the problem definition.	Student has not understood the concepts and the problem definition clearly.
Design of program [CO2-6] [PO2, PO3]	Capable of discussing multiple designs and selecting the best with justification.	Discusses multiple designs but lacks proper selection criteria.	Discusses a single design with merits and.	Explains only the basic design.	Unable to explain the design.
Implementation using Java [CO2-6] [PO3, PO8, PO5]	Implements optimal solution using suitable Java constructs and OOP features.	Correct implementation with good explanation. demerits	Correct implementation with partial explanation.	Implementation contains errors.	Unable to implement the solution.
Program debugging and testing with suitable tools [CO2-6] [PO5, PO8]	Compiles, debugs, and tests without errors; demonstrates full understanding.	Corrects errors with good understanding.	Corrects errors with partial understanding.	Corrects errors with assistance.	Unable to debug and test independently.
Results & interpretation /analysis [CO1-6] [PO4, PO5]	Executes for multiple test cases and performs detailed analysis.	Executes correctly for all test cases.	Executes for a few test cases and analyzes results.	Executes but cannot analyze results.	Unable to verify correctness of results.
Demonstration and documentation	Demonstration and lab record	Demonstration and lab record is	Demonstration and lab record	Demonstration and lab record is	Demonstration and lab record is poorly

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - CYBER SECURITY

[CO1-6] [PO8, PO9, PO11]	is well-organized, with clear sections. The record is well structured with suitable formatting.	organized, with clear sections, but some sections are not well-defined. The record is structured with formatting.	lacks clear organization or structure. Some sections are unclear or incomplete. The record is partially structured with formatting.	poorly organized, with missing or unclear sections. The record is not properly structured with suitable formatting.	organized, with missing sections. Record not submitted on time. The record is not structured with minimum formatting.
-----------------------------	---	---	---	---	---

Rubrics for CIE Practical Test:

Criteria/ Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Unacceptable (1)
Fundamental Knowledge (2) [CO1] [PO1]	The student has well depth knowledge of the topics related to the problem & course.	Student has good knowledge of some of the topics related to problem & course.	Student has average knowledge of some of the topics related to problem & course.	Student is capable of narrating the answer but not capable to show in depth knowledge.	Student has not understood the concepts clearly.
Understanding of problem definition (1) [CO2-6] [PO2]	Student is able to completely understand the problem definition.	Student is able to understand the problem definition but not clearly.	Student has a basic understanding of the problem definition that is partial or superficial.	Student is able to show minimal or unclear understanding of the problem definition.	Student is not able to understand the problem definition.
Design and Implementation (3) [CO2-6] [PO3, PO5]	Student is capable of design and implementing with best suitable java construct for the given problem definition.	Student is capable of design and implementing with some construct for the given problem definition.	Student is capable of design and implementing the core part of the construct for the given problem definition.	Student is partially capable of design and implementing with some algorithm for the given problem definition.	Student is not capable of design and implementing.
Result & Analysis (2) [CO2-6] [PO4, PO5]	Student is able to run the program on various data inputs and compare the result with proper inference.	Student will be able to run the program on various data inputs and fair knowledge in comparing the result with proper inference.	Student will be able to run the code for few data/datasets and analyze the output.	Student will be able to run the code for few data inputs but not analyze the output.	Student will be not able to run the program and not able to analyze the result.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - CYBER SECURITY

Communication (Viva voce) (2) [CO1-6] [PO8, PO9]	Good Verbal & nonverbal communication skills with precise and correct terminologies/ answers.	Good verbal Communication skills with precise and correct terminologies/ans wers.	Average Communication but with precise and correct terminologies/ans wers.	Average Communication but with imprecise and incorrect terminologies/ans wers.	Poor Communication (Minimal interaction/ans wers)
---	--	--	---	---	---

Term Work (TW) and Self Learning (SL) components in Number of hours per semester		
Term work (includes assignments, seminars, micro projects, industrial visits, any other student activities, etc.)		
Sl. No.	Term Work (TW) Activity	Number of Hours / Semester

1.	Installation and use of IDE for Java. Writing simple Java programs to understand Java program development and execution model.	04
SL: Self Learning, MOOCs, spoken tutorials, online educational resources etc.		
Sl. No.	Self-Learning (SL) Activity	Number of Hours / Semester
1.	Springboard course on: Java Programming Fundamentals	36
2	Solve problems on HackerRank using Java programming https://www.hackerrank.com/domains/java	10

Course Code:	1BC302	TITLE: OBJECT ORIENTED PROGRAMMING WITH JAVA						Faculty Name:		Dr.Shwetha G K		
List of Course Outcomes	Program Outcomes											Total
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	
CO-1	2	2		-	-	-	-	-	-	-	-	4
CO-2	2	2		-	-	-	-	-	-	-	-	4
CO-3	3	3	2	-	-	-	-	-	-	-	-	8
CO-4	3	3	2	-	-	-	-	-	-	-	-	8
CO-5	2	3		-	-	-	-	-	-	-	-	5
Total	12	13	4	-	-	-	-	-	-	-	-	29

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - CYBER SECURITY

The Correlation of Course Outcomes (CO's) and Program Specific Outcomes (PSO's)

Note: 3 = Strong Contribution, 2 = Average Contribution, 1 = Weak Contribution, - = No Contribution

Course Code:	1BCS302	TITLE: OBJECT ORIENTED PROGRAMMING WITH JAVA	Faculty Name:	
List of Course Outcomes	Program Specific Outcomes			Total
	PSO-1	PSO-2	PSO-3	
CO-1	-	2	-	2
CO-2	-	2	-	2
CO-3	-	2	-	2
CO-4	-	2	-	2
CO-5	-	2	-	2
Total	-	2	-	10

Course Coordinator

HODs



ATME

College of Engineering



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - CYBER SECURITY

