INFORMATION SECURITY		Semester	4
Course Code	**BCR701**	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:2:0	SEE Marks	50
Total Hours of Pedagogy	40 hours Theory + 8-10 Lab slots	Total Marks	100
Credits	04	Exam Hours	03
Examination nature (SEE)	Theory/Practical		

Course objectives:

- Understand the basics of Cryptography concepts, Security and its principle
- To analyse different Cryptographic Algorithms
- To illustrate public and private key cryptography
- To understand the key distribution scenario and certification
- To understand approaches and techniques to build protection mechanism in order to secure computer networks

Teaching-Learning Process (General Instructions)

Teaching and learning process can be defined as a transformation process of knowledge from teachers to students. It is referred as the combination of various elements within the process where an educator identifies and establish the learning objectives and develop teaching resources and implement the teaching and learning strategy. On the other hand, learning is a cardinal factor that a teacher must consider while teaching students.

MODULE-1	**No. of Hours: 8**

A model for Network Security, Classical encryption techniques: Symmetric cipher model, Substitution ciphers-Caesar Cipher, Monoalphabetic Cipher, Playfair Cipher, Hill Cipher, Polyalphabetic Ciphers, One time pad, Steganography

Block Ciphers and Data Encryption Standards: Traditional Block Cipher structures, data Encryption Standard (DES), A DES Example, The strength of DES, Block cipher design principles

Chapter 1: 1.8 Chapter 3: 3.1, 3.2, 3.5 Chapter 4: 4.1, 4.2, 4.3, 4.4, 4.5

MODULE-2	**No. of Hours: 8**

Pseudorandom number Generators: Linear Congruential Generators, Blum Blum Shub Generator

Public key cryptography and RSA: Principles of public key cryptosystems-Public key cryptosystems, Applications for public key cryptosystems, Requirements for public key cryptography, Public key Cryptanalysis, The RSA algorithm: Description of the Algorithm, Computational aspects, The Security of RSA

Diffie-Hellman key exchange: The Algorithm, Key exchange Protocols, Man-in-the-middle Attack, Elliptic Curve Cryptography: Analog of Diffie-Hellman key Exchange, Elliptic Curve Encryption/Decryption, Security of Elliptic Curve Cryptography

Chapter 8: 8.2 Chapter 9: 9.1, 9.2 Chapter 10: 10.1, 10.4

MODULE-3	**No. of Hours:8**

Applications of Cryptographic Hash functions, Two simple Hash functions, Key management and distribution: Symmetric key distribution using symmetric encryption, Symmetric key distribution using asymmetric encryption, Distribution of public keys, X.509 Certificates, Public Key Infrastructures

Chapter 11: 11.1, 11.2 Chapter 14: 14.1, 14.2, 14.3, 14.4, 14.5

MODULE-4	**No. of Hours:8**

User Authentication: Remote user authentication principles**,** Kerberos**,** Remote user authentication using asymmetric encryption Web security consideration, Transport layer security Email Threats and comprehensive email security, S/MIME, Pretty Good Privacy Chapter 15: 15.1, 15.3, 15.4 Chapter 17: 17.1, 17.2 Chapter 19: 19.3, 19.4, 19.5

MODULE-5	**No. of Hours:08**
Domain keys Identified Mail IP Security: IP Security overview, IP Security Policy, Encapsulating Security Payload, Combining security associations, Internet key exchange Chapter 19: 19.9 Chapter 20: 20.1, 20.2, 20.3, 20.4, 20.5	

PRACTICAL COMPONENT OF IPCC

Sl.NO	Experiments (Implement using C/C++/Java Programming Languages)
1	Implement Caesar Cipher substitution method
2	Implement Mono alphabetic cipher with the given key and input
3	Implement Poly alphabetic Cipher
4	Encrypt the given text using Play fair Cipher with the given key. Also perform the decryption operation
5	Implement Encryption and Decryption techniques in Hill Cipher
6	Demonstrate Single and Double Transposition techniques
7	Implement Simple DES/DES Algorithm
8	Generate Pseudo random numbers using Linear Congruential method
9	Generate Pseudo random numbers using Blum Blum Shub Generator
10	Implement RSA Algorithm
11	Demonstrate Diffie Hellman Key exchange Algorithm
12	Implement Fermat's and Euler's Theorem (Course Instructor need to explain the theorem before execution as the topic not covered in Theory part)

Course outcome

At the end of the course, the student will be able to :

CO1: Explain the basic concepts of Cryptography and Security aspects

CO2: Apply different Cryptographic Algorithms for different applications

CO**3:** Analyze different methods for authentication and access control.

CO4: Describe key management, key distribution and Certificates.

CO5: Explain about Electronic mail and IP Security.

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE

minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

CIE for the theory component of the IPCC (maximum marks 50)

- IPCC means practical portion integrated with the theory of the course.
- CIE marks for the theory component are **25 marks** and that for the practical component is **25 marks**.
- 25 marks for the theory component are split into **15 marks** for two Internal Assessment Tests (Two Tests, each of 15 Marks with 01-hour duration, are to be conducted) and **10 marks** for other assessment methods mentioned in 22OB4.2. The first test at the end of 40-50% coverage of the syllabus and the second test after covering 85-90% of the syllabus.
- Scaled-down marks of the sum of two tests and other assessment methods will be CIE marks for the theory component of IPCC (that is for **25 marks)**.
- The student has to secure 40% of 25 marks to qualify in the CIE of the theory component of IPCC.

CIE for the practical component of the IPCC

- **15 marks** for the conduction of the experiment and preparation of laboratory record, and **10 marks** for the test to be conducted after the completion of all the laboratory sessions.
- On completion of every experiment/program in the laboratory, the students shall be evaluated including viva-voce and marks shall be awarded on the same day.
- The CIE marks awarded in the case of the Practical component shall be based on the continuous evaluation of the laboratory report. Each experiment report can be evaluated for 10 marks. Marks of all experiments' write-ups are added and scaled down to **15 marks**.
- The laboratory test **(duration 02/03 hours)** after completion of all the experiments shall be conducted for 50 marks and scaled down to **10 marks.**
- Scaled-down marks of write-up evaluations and tests added will be CIE marks for the laboratory component of IPCC for **25 marks**.
- The student has to secure 40% of 25 marks to qualify in the CIE of the practical component of the IPCC.

SEE for IPCC

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the course (**duration 03 hours**)

1. The question paper will have ten questions. Each question is set for 20 marks.
2. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), **should have a mix of topics** under that module.
3. The students have to answer 5 full questions, selecting one full question from each module.
4. Marks scoredby the student shall be proportionally scaled down to 50 Marks

The theory portion of the IPCC shall be for both CIE and SEE, whereas the practical portion will have a CIE component only. Questions mentioned in the SEE paper may include questions from the practical component.

Suggested Learning Resources:

Text Book:

William Stallings, "Cryptography and Network Security", Pearson Publication, Seventh Edition.

Reference Books:

1. Keith M Martin, "Everyday Cryptography", Oxford University Press
2. V.K Pachghare,"Cryptography and Network Security", PHI, 2nd Edition
3. Everyday Cryptography: Fundamental Principles and Applications Keith M. Martin

Oxford Scholarship Online: December 2013

4. Information Security: Principles and Practice, 2nd Edition by Mark Stamp Wiley

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- Group assignment to implement Cryptographic Algorithms (Not covered in Laboratory experiment list) – 10 Marks

PARALLEL COMPUTING		Semester	VII
Course Code	**BCS702**	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:2:0	SEE Marks	50
Total Hours of Pedagogy	40 hours Theory + 8-10 Lab slots	Total Marks	100
Credits	04	Exam Hours	03
Examination nature (SEE)	Theory/Practical		

Course objectives:

This course will enable to,

- Explore the need for parallel programming
- Explain how to parallelize on MIMD systems
- To demonstrate how to apply MPI library and parallelize the suitable programs
- To demonstrate how to apply OpenMP pragma and directives to parallelize the suitable programs
- To demonstrate how to design CUDA program

Teaching-Learning Process (General Instructions)

These are sample Strategies that teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) need not to be only traditional lecture methods, but alternative effective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Programming assignment, which fosters student's Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.

MODULE-1

Introduction to parallel programming, Parallel hardware and parallel software – Classifications of parallel computers, SIMD systems, MIMD systems, Interconnection networks, Cache coherence, Shared-memory vs. distributed-memory, Coordinating the processes/threads, Shared-memory, Distributed-memory.

MODULE-2

GPU programming, Programming hybrid systems, MIMD systems, GPUs, Performance – Speedup and efficiency in MIMD systems, Amdahl's law, Scalability in MIMD systems, Taking timings of MIMD programs, GPU performance.

MODULE-3

Distributed memory programming with MPI – MPI functions, The trapezoidal rule in MPI, Dealing with I/O, Collective communication, MPI-derived datatypes, Performance evaluation of MPI programs, A parallel sorting algorithm.

MODULE-4

Shared-memory programming with OpenMP – openmp pragmas and directives, The trapezoidal rule, Scope of variables, The reduction clause, loop carried dependency, scheduling, producers and consumers, Caches, cache coherence and false sharing in openmp, tasking, tasking, thread safety.

MODULE-5

GPU programming with CUDA - GPUs and GPGPU, GPU architectures, Heterogeneous computing, Threads, blocks, and grids Nvidia compute capabilities and device architectures, Vector addition, Returning results from CUDA kernels, CUDA trapezoidal rule I, CUDA trapezoidal rule II: improving performance, CUDA trapezoidal rule III: blocks with more than one warp.

PRACTICAL COMPONENT OF IPCC

Sl.NO	Experiments
1	Write a OpenMP program to sort an array on n elements using both sequential and parallel mergesort(using Section). Record the difference in execution time.
2	Write an OpenMP program that divides the Iterations into chunks containing 2 iterations, respectively (OMP_SCHEDULE=static,2). Its input should be the number of iterations, and its output should be which iterations of a parallelized for loop are executed by which thread. For example, if there are two threads and four iterations, the output might be the following: a. Thread 0 : Iterations 0 −− 1 b. Thread 1 : Iterations 2 −− 3
3	Write a OpenMP program to calculate n Fibonacci numbers using tasks.
4	Write a OpenMP program to find the prime numbers from 1 to n employing parallel for directive. Record both serial and parallel execution times.
5	Write a MPI Program to demonstration of MPI_Send and MPI_Recv.
6	Write a MPI program to demonstration of deadlock using point to point communication and avoidance of deadlock by altering the call sequence
7	Write a MPI Program to demonstration of Broadcast operation.
8	Write a MPI Program demonstration of MPI_Scatter and MPI_Gather
9	Write a MPI Program to demonstration of MPI_Reduce and MPI_Allreduce (MPI_MAX, MPI_MIN, MPI_SUM, MPI_PROD)

Course outcomes (Course Skill Set):

At the end of the course, the student will be able to:

- Explain the need for parallel programming
- Demonstrate parallelism in MIMD system.
- Apply MPI library to parallelize the code to solve the given problem.
- Apply OpenMP pragma and directives to parallelize the code to solve the given problem
- Design a CUDA program for the given problem.

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

CIE for the theory component of the IPCC (maximum marks 50)

- IPCC means practical portion integrated with the theory of the course.
- CIE marks for the theory component are **25 marks** and that for the practical component is **25 marks**.
- 25 marks for the theory component are split into **15 marks** for two Internal Assessment Tests (Two Tests, each of 15 Marks with 01-hour duration, are to be conducted) and **10 marks** for other assessment methods mentioned in 22OB4.2. The first test at the end of 40-50% coverage of the syllabus and the second test after covering 85-90% of the syllabus.
- Scaled-down marks of the sum of two tests and other assessment methods will be CIE marks for the theory component of IPCC (that is for **25 marks)**.
- The student has to secure 40% of 25 marks to qualify in the CIE of the theory component of IPCC.

CIE for the practical component of the IPCC

- **15 marks** for the conduction of the experiment and preparation of laboratory record, and **10 marks** for the test to be conducted after the completion of all the laboratory sessions.
- On completion of every experiment/program in the laboratory, the students shall be evaluated including viva-voce and marks shall be awarded on the same day.
- The CIE marks awarded in the case of the Practical component shall be based on the continuous evaluation of the laboratory report. Each experiment report can be evaluated for 10 marks. Marks of all experiments' write-ups are added and scaled down to **15 marks**.
- The laboratory test **(duration 02/03 hours)** after completion of all the experiments shall be conducted for 50 marks and scaled down to **10 marks.**
- Scaled-down marks of write-up evaluations and tests added will be CIE marks for the laboratory component of IPCC for **25 marks**.
- The student has to secure 40% of 25 marks to qualify in the CIE of the practical component of the IPCC.

SEE for IPCC

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the course (**duration 03 hours**)

1. The question paper will have ten questions. Each question is set for 20 marks.
2. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), **should have a mix of topics** under that module.
3. The students have to answer 5 full questions, selecting one full question from each module.
4. Marks scored by the student shall be proportionally scaled down to 50 Marks

The theory portion of the IPCC shall be for both CIE and SEE, whereas the practical portion will have a CIE component only. Questions mentioned in the SEE paper may include questions from the practical component.

Suggested Learning Resources:

Textbook:

1. Peter S Pacheco, Matthew Malensek – An Introduction to Parallel Programming, second

edition, Morgan Kauffman.
2. Michael J Quinn – Parallel Programming in C with MPI and OpenMp, McGrawHill.

Reference Books:

1. Calvin Lin, Lawrence Snyder – Principles of Parallel Programming, Pearson
2. Barbara Chapman – Using OpenMP: Portable Shared Memory Parallel Programming, Scientific and Engineering Computation
3. William Gropp, Ewing Lusk – Using MPI:Portable Parallel Programing, Third edition, Scientific and Engineering Computation

Web links and Video Lectures (e-Resources):

1. Introduction to parallel programming: https://nptel.ac.in/courses/106102163

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- Programming Assignment at higher bloom level (10 Marks)

Principles of Programming Languages		Semester	7
Course Code	BCR703	CIE Marks	50
Teaching Hours/Week (L: T:P: S)	4:0:0:0	SEE Marks	50
Total Hours of Pedagogy	50	Total Marks	100
Credits	04	Exam Hours	03
Examination type (SEE)	Theory		

Course objectives:
This course will enable students to,

- Learn the data types, binding types and scope in programming languages.
- Understand the role of expression, statements and control instructions in programing.
- Gain knowledge on functions, enumerations and abstract data types.

Teaching-Learning Process (General Instructions)
These are sample Strategies; that teachers can use to accelerate the attainment of the various course outcomes.

1. Lecture method (L) need not be only a traditional lecture method; alternative effective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain the functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher Order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Problem Based-Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than recall it.

Module-1

Preliminaries: Reasons for Studying Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-Offs, Implementation Methods, Programming Environments.
Names, Bindings, and Scopes: Introduction, Names, Variables, The Concept of Binding, Scope, Scope and Lifetime, Referencing Environments, Named Constants.

Chapter 1 (1.1-1.8), Chapter 5 (5.1-5.8)

Module-2

Data Types: Introduction, Primitive Data Types, Character String Types, Enumeration Types, Array Types, Associative Arrays, Record Types, Tuple Types, List Types, Union Types, Pointer and Reference Types, Optional Types, Type Checking, Strong Typing, Type Equivalence.

Chapter 6 (6.1-6.15)

Module-3

Expressions and Assignment Statements: Introduction, Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation, Assignment Statements, Mixed-Mode Assignment.
Statement-Level Control Structures: Introduction, Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands.

Chapter 7 (7.1-7.8), Chapter 8 (8.1-8.5)

Module-4

Subprograms: Introduction, Fundamentals of Subprograms, Design Issues for Subprograms, Local Referencing Environments, Parameter-Passing Methods, Parameters That Are Subprograms, Calling Subprograms Indirectly, Design Issues for Functions, Overloaded Subprograms, Generic Subprograms, User-Defined Overloaded Operators, Closures, Co-routines.

Chapter 9 (9.1-9.13)

Module-5

Implementing Subprograms: The General Semantics of Calls and Returns, Implementing "Simple" Subprograms, Implementing Subprograms with Stack-Dynamic Local, Variables, Nested Subprograms, Blocks, Implementing Dynamic Scoping.

Abstract Data Types and Encapsulation Constructs: The Concept of Abstraction, Introduction to Data Abstraction, Design Issues for Abstract Data Types, Parameterized Abstract Data Types, Encapsulation Constructs, Naming Encapsulations.

Chapter 10 (10.1-10.6), Chapter 11(11.1-11.3, 11.5-11.7)

Course outcome (Course Skill Set)

At the end of the course, the student will be able to :

1. Explain names, binding types, and scope rules in programming languages.
2. Describe different data types and their significance in programming.
3. Classify expressions, assignments, and control structures.
4. Explain the role of subprograms in program development.
5. Model abstraction, abstract data types, and encapsulation features.

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

Continuous Internal Evaluation:

- For the Assignment component of the CIE, there are 25 marks and for the Internal Assessment Test component, there are 25 marks.
- The first test will be administered after 40-50% of the syllabus has been covered, and the second test will be administered after 85-90% of the syllabus has been covered
- Any two assignment methods mentioned in the 22OB2.4, if an assignment is project-based then only one assignment for the course shall be planned. The teacher should not conduct two assignments at the end of the semester if two assignments are planned.
- For the course, CIE marks will be based on a scaled-down sum of two tests and other methods of assessment.

Internal Assessment Test question paper is designed to attain the different levels of Bloom's taxonomy as per the outcome defined for the course.

Semester-End Examination:

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the course (**duration 03 hours).**

1. The question paper will have ten questions. Each question is set for 20 marks.
2. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), **should have a mix of topics** under that module.
3. The students have to answer 5 full questions, selecting one full question from each module.
4. Marks scored shall be proportionally reduced to 50 marks

Suggested Learning Resources:

Textbook:

1. Robert W Sebesta, CONCEPTS OF PROGRAMMING LANGUAGES, 12th Global Edition, Pearson, 2022.

Reference Books:

1. Terrance W Pratt, Marvin V Z and T V Gopal, Programming Languages – Design and Implementation, 4e, Pearson, 2006.
2. Michael Scott, Programming Language Pragmatics, Third Edition, Morgan Kaufmann Publishers (Elsevier), 2009.

Web links and Video Lectures (e-Resources):

- https://nptel.ac.in/courses/106102067
- https://pl.cs.jhu.edu/pl/book/book.pdf
- https://www.cs.toronto.edu/~david/course-notes/csc324.pdf

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- Programming assignments (two) related to language constructs/features described in the syllabus (15 & 10 marks)