

 Estd. 1962 "A" Accredited by NAAC(2021) With CGPA 3.52	<b>SHIVAJI UNIVERSITY, KOLHAPUR - 416004,</b> <b>MAHARASHTRA</b> PHONE : EPABX – 2609000, <a href="http://www.unishivaji.ac.in">www.unishivaji.ac.in</a> , <a href="mailto:bos@unishivaji.ac.in">bos@unishivaji.ac.in</a> <b>शिवाजी विद्यापीठ, कोल्हापूर - ४१६००४, महाराष्ट्र</b> दूरध्वनी - ईपीएबीएक्स - २६०९०००, अभ्यासमंडळे विभाग दूरध्वनी विभाग २३१-२६०९०९३/९४	
--	---	---

SU/BOS/Science/

No 00323 Date: 19 SEP 2022

To,  
 The Principal,  
 All Affiliated Concerned Science Colleges/Institutions  
 Shivaji University, Kolhapur.

**Subject :- Regarding syllabi of M.Sc., MCA & BCA (NEP-2020) degree programme under the Faculty of Science and Technology as per National Education Policy, 2020 .**

Sir/Madam,

With reference to the subject mentioned above, I am directed to inform you that the university authorities have accepted and granted approval to the syllabi and Nature of question paper of **M.Sc., MCA & BCA** under the Faculty of Science and Technology as per National Education Policy, 2020 .

Sr. No.	Faculty of Science and Technology	Programme/ Course
1	<b>Computer Science Engineering and Technology</b>	M. Sc. Part- I Computer Science
		MCA. Part- I Computer Science
		BCA. Part- I Bachelor of Computer Application

This syllabi and nature of question paper shall be implemented from the Academic Year 2022-2023 onwards. A soft copy containing the syllabus is attached herewith and it is also available on university website [www.unishivaji.ac.in](http://www.unishivaji.ac.in) (students Online Syllabus)

You are, therefore, requested to bring this to the notice of all students and teachers concerned.

Thanking you,

Yours faithfully,

Dy Registrar

Copy to:

1	The Dean, Faculty of Science & Technology	7	Appointment Section
2	Director, Board of Examinations and Evaluation	8	P.G.Seminar Section
3	The Chairman, Respective Board of Studies	9	Computer Centre (I.T.)
4	B.Sc. Exam	10	Affiliation Section (U.G.)
5	Eligibility Section	11	Affiliation Section (P.G.)
6	O.E. I Section	12	P.G.Admission Section

# **SHIVAJI UNIVERSITY, KOLHAPUR**



NAAC “A++” Grade with CGPA 3.52

**Choice Based Credit System with Multiple Entry and Multiple Exit Option (NEP-2020)**

**Syllabus for**

**Master of Science**

**In**

**Computer Science**

(Under Faculty of Science and Technology)

**PART I SEMESTER I & II**

**(Syllabus to be implemented from Academic year 2022-23)**

**Choice Based Credit System with Multiple Entry and Multiple Exit Option (NEP-2020)**  
**M.Sc. Program Structure**  
**M.Sc. Part – I (Level-8)**

<b>SEMESTER-I (Duration- Six Month)</b>											
	Sr. No.	CourseCode	Teaching Scheme			Examination Scheme					
			Theory and Practical			University Assessment (UA)			Internal Assessment (IA)		
			Lectures (Per week)	Hours (Per week)	Credit	Maximum Marks	Minimum Marks	Exam. Hours	Maximum Marks	Minimum Marks	Exam. Hours
<b>CGPA</b>	1	CC-101: Design and Analysis of Algorithm	4	4	4	80	32	3	20	8	1
	2	CC-102: Python Programming	4	4	4	80	32	3	20	8	1
	3	CC-103: Database Management System	4	4	4	80	32	3	20	8	1
	4	CC-104: Cyber Security	4	4	4	80	32	3	20	8	1
	5	CCPR-105: Python Lab	-	6	4	100	40	3	--	--	*
	6	CCPR-106: Database Lab	-	6	4	100	40	3	--	--	*
<b>Total (A)</b>			<b>16</b>	<b>28</b>	<b>24</b>	<b>520</b>	<b>--</b>	<b>--</b>	<b>80</b>	<b>--</b>	<b>--</b>
<b>Non- CGPA</b>	1	AEC-107: Communicative English-I	2	2	2	--	--	--	50	20	2
<b>SEMESTER-II (Duration- Six Month)</b>											
<b>CGPA</b>	1	CC-201: Web Technology	4	4	4	80	32	3	20	8	1
	2	CC-202: Advanced Java	4	4	4	80	32	3	20	8	1
	3	CC-203: Android development with Kotlin	4	4	4	80	32	3	20	8	1
	4	CCS-204: 1.Software Project Management	4	4	4	80	32	3	20	8	1

		2.Data Science Foundation 3.Application Security Analyst 4.Cloud Computing									
	5	CCPR-205: Web Technology Lab and Advanced Java Lab	-	6	4	100	40	3	--	--	*
	6	CCPR-206: Project	-	6	4	100	40	3	--	--	*
<b>Total (B)</b>			--	--	<b>24</b>	<b>520</b>	--	--	<b>80</b>	--	--
<b>Non-CGPA</b>	1	SEC-207	2	2	2	--	--	--	50	20	2
<b>Total (A+B)</b>					<b>48</b>	<b>1040</b>	--	--	<b>160</b>	--	--

<ul style="list-style-type: none"> <li>Student contact hours per week : <b>56</b> Hours (Min.)</li> <li>Theory and Practical Lectures : <b>60</b> Minutes Each</li> <li>CC-Core Course</li> <li>CCPR-Core Course Practical</li> <li>AEC-Mandatory Non-CGPA compulsory Ability Enhancement Course</li> <li>SEC- Mandatory Non-CGPA compulsory Skill Enhancement Course</li> </ul>	<ul style="list-style-type: none"> <li>Total Marks for M.Sc.-I : <b>1040</b></li> <li>Total Credits for M.Sc.-I (Semester I &amp; II) : <b>48</b></li> <li><b>Practical Examination is Semester wise after theory examination.</b></li> <li>Examination for CCPR-105and CCPR-106 shall be based on Semester I Practical.</li> <li>Examination for CCPR-205 and CCPR-206 shall be based on Semester II Practical.</li> <li>*Duration of Practical Examination as per respective BOS guidelines</li> <li><i>Separate passing is mandatory for Theory, Internal and Practical Examination</i></li> </ul>
<ul style="list-style-type: none"> <li><b>Requirement for Entry at Level 8:</b> <b>Completed all requirements of the relevant Bachelor's degree (Level 7) .</b></li> </ul>	
<ul style="list-style-type: none"> <li><b>Exit Option at Level 8:</b> Students can exit after Level 8 with <b>Post Graduate Diploma in Computer Science</b> if he/she completes the courses equivalent to minimum of 48 credits.</li> </ul>	

**Choice Based Credit System with Multiple Entry and Multiple Exit Option (NEP-2020)**  
**M.Sc.. Program Structure**  
**M.Sc. Part – II (Level-9)**

SEMESTER-III (Duration- Six Month)											
	Sr. No.	CourseCode	Teaching Scheme			Examination Scheme					
			Theory and Practical			University Assessment (UA)			Internal Assessment (IA)		
			Lectures (Per week)	Hours (Per week)	Credit	Maximum Marks	Minimum Marks	Exam. Hours	Maximum Marks	Minimum Marks	Exam. Hours
CGPA	1	CC-301:Artificial Intelligence	4	4	4	80	32	3	20	8	1
	2	CC-302: Advanced Web Technology	4	4	4	80	32	3	20	8	1
	3	CC -303:PHP	4	4	4	80	32	3	20	8	1
	4	CCS-304: 1.Software Quality Assurance 2.Advance Data Science 3.Network Security Analyst 4.Internet of Things	4	4	4	80	32	3	20	8	1
	5	CCPR-305: Advanced Web Technology Lab	--	6	4	100	40	3	--	--	*
	6	CCPR-306: PHP Lab	--	6	4	100	40	3	--	--	*
	6	CCPR-307: Project	--	6	4	100	40	3	--	--	*
Total (C)			--	--	28	620	--	--	80	--	
Non-CGPA	1	AEC-308: Communicative English-II	2	2	2	--	--	--	50	20	2
	2	EC(SWMMOOC)-309:	Number of Lectures and credit shall be specified on SWAYAM MOOC								
SEMESTER-IV (Duration- Six Month)											
CGPA	1	CCPR-401:Research Seminar		1	4				100	40	*
	6	CCPR-402: Research /Industrial		-	16	300	120	--	100	40	*

		Project								
		<b>Total (D)</b>	--	<b>1</b>	<b>20</b>	<b>300</b>	--	--	<b>200</b>	--
	1	GE-403:	2	2	2	--	--	--	50	20
		<b>Total (C+D)</b>			<b>40</b>	<b>920</b>	--	--	<b>280</b>	--

<ul style="list-style-type: none"> <li>• Student contact hours per week : <b>34</b> Hours (Min.)</li> </ul>	<ul style="list-style-type: none"> <li>• Total Marks for MSc.-II : <b>1200</b></li> </ul>
<ul style="list-style-type: none"> <li>• Theory and Practical Lectures : <b>60</b> Minutes Each</li> </ul>	<ul style="list-style-type: none"> <li>• Total Credits for MSc.-II (Semester III &amp; IV) : <b>48</b></li> </ul>
<ul style="list-style-type: none"> <li>• CC-Core Course</li> <li>• CCS- Core Course Specialization</li> <li>• CCPR-Core Course Practical and Project</li> <li>• DSE-Discipline Specific Elective</li> <li>• AEC-Mandatory Non-CGPA compulsory Ability Enhancement Course</li> <li>• SEC- Mandatory Non-CGPA compulsory Skill Enhancement Course</li> <li>• EC (SWM MOOC) - Non-CGPA Elective Course</li> <li>• GE- Multidisciplinary Generic Elective</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Practical Examination is Semester wise after theory examination.</b> Examination for CCPR-305 and CCPR-306 shall be based on Semester III Practical.</li> <li>• Examination for CCPR-401 and CCPR-402 shall be based on Semester IV Practical.</li> <li>• *Duration of Practical Examination as per respective BOS guidelines</li> <li>• <b><i>Separate passing is mandatory for Theory, Internal and Practical Examination</i></b></li> </ul>
<ul style="list-style-type: none"> <li>• <b>Requirement for Entry at Level 9:</b> <b>Completed all requirements of the relevant Post Graduate Diploma in Computer Science (Level 8)</b></li> </ul>	
<ul style="list-style-type: none"> <li>• <b>Exit at Level 9:</b> Students will exit after Level 9 with <b>Master's Degree in Computer Science</b> if he/she completes the courses equivalent to minimum of 96 credits.</li> </ul>	

	<b>M.Sc. I</b>	<b>M.Sc. II</b>	<b>Total</b>
<b>Marks</b>	<b>1200</b>	<b>1200</b>	<b>2400</b>
<b>Credits</b>	<b>48</b>	<b>48</b>	<b>96</b>

The name of the programme shall be Master of Science (**M.Sc. in Computer Science**)

### **Duration of the Program**

- The M.Sc. programme will be a full-time two years i.e. 4 semesters. Pattern of examination will be Semester System.

### **Medium of Instruction**

- The medium of Instruction will be English only.

### **Admission Procedure**

- Eligibility: B.Sc. Computer Science (Entire/ optional) / B.Sc. IT/ BCA, B.Sc. Mathematics, B.Sc. Statistics, B.Sc. Electronics, B.Sc Animation, B.Sc Physics, B.Sc Chemistry/ BSc. Microbiology.
- Admission through University Entrance exam only.
- Only entrance marks should be considered for admission process.
- Reservation of Seats as per rules of Government of Maharashtra.

### **Project work**

**At the end of I, II & III semester student has to carry out a project work.**

1. Project work at end of semester should be done in groups, each student must be given a responsibility for a distinct module and care should be taken to see the progress of individual modules is independent of others.
2. Students should take guidance from an internal guide and prepare a Project Report on "Project Work" to be submitted to the Department after evaluation.
3. The Project Report should contain an Introduction to Project, which should clearly explain the project scope in detail. Database/Webpage/ UI designs and a list of output reports should be included along with references.
4. The project Work should be of such a nature that it could prove useful or should be relevant from the societal/commercial/research angle.
5. The project report will be duly accessed by the internal guide of the project and internal marks will be communicated by the concerned guide.
6. Project viva-voce by the University panel will be conducted as part of Evaluation.

**At the end of the Fourth semester of study, a student will be examined in the course "Industrial / Research Project ".**

1. Fourth semester Project work can be carried out as industrial training of four months in the Industry or in the Institute as Research project with prior permission of the Institute.
2. Project viva-voce by the University panel will be conducted at the end of semester.
3. The project report should be prepared in a format prescribed by the University, which also specifies the contents and methods of presentation.
4. Project work may be done individually or in groups in case of bigger projects.



5. The major project work carry 50 marks for internal assessment and 150 marks for External viva. The external viva shall be conducted by a panel of external examiners.

### **OR**

1. The student will be allowed to formulate a proposal for start-up and the same will be rated equivalent to an industrial project. A detailed problem statement showing innovation along with markability, business plan and cash flow will be part of the Evaluation criteria.

### **Research Seminar**

At the end of fourth semester student shall deliver seminar on one of the advanced topic chosen in consultation with the guide after compiling the information from the latest literature and also internet. The concepts must be clearly understood and presented by student. Prior to presentation, he/she shall carry out the detailed literature survey from standard references such as International & National journals and periodicals recently published reference books etc. A hard copy of the report (A4 size, 12 fonts, Times New Roman, Single spacing both side printed) should be submitted to the Department before delivering the seminar. This seminar will be evaluated internally for 100 marks by the respective guides.

### **Assessment**

The final total assessment of the candidate is made in terms of an internal assessment and an external assessment for each course.

1. For each theory paper, 20% marks will be based on internal assessment and 80% marks for semester examination (external assessment), unless otherwise stated.
2. Internal assessment of theory papers should be in the form of two internal tests of 10 marks each. Total 20 marks.
3. The projects will be evaluated by the university appointed panel.
4. The final practical examination will be conducted by the university appointed panel at the end of semester for each lab course and marks will be submitted to the university by the panel. The pattern of final Practical Examination will be as follows-

1	Coding and Execution of Program	60 Marks
2	Viva-voce	20 Marks
3	Journal	20 Marks
	Total	100 marks

6. The internal marks will be communicated to the University at the end of each semester, but before the semester end examinations. These marks will be considered for the declaration of the results.

#### **Nature of question paper**

Nature of question paper is as follows for University end semester examination

a. **Theory Examination:** There will be seven (7) questions of 16 Marks and out of which four (4) to be attempted from question no 2 to 6. Question No.1 is compulsory and is of multiple choice questions.

#### **b. Practical Examination:**

i. Duration of Practical Examination: 3 Hrs

ii. Nature of Question paper: There will be three questions out of which any two questions to be attempted and each question carries 30 Marks.

#### **Standard of Passing**

Internal as well as external examination will be held at the end of semester. The candidate must score 40% marks in each head of internal as well as external Examination.

#### **Board of Paper Setters /Examiners**

For each Semester and examination there will be one board of Paper setters and examiners for every course.

#### **Award of Class**

There will be numerical marking on each question. At the time of declaration of the result the marks obtained by the candidate is converted into classes as per University norms.

### **Credit system implementation**

As per the University norms

### **Clarification of Syllabus**

The syllabus Committee should meet at least once in a year to study and clarify any difficulties from the Institutes. The Workshop on syllabi should be organised at the beginning of every semester.

### **Revision of Syllabus**

As the computer technology is changing very fast, revision of the syllabus should be considered every 3 years.

### **PEO's for M.Sc. Program**

Program Educational Objectives is to prepare graduates to:

1. Apply and continuously acquire knowledge, both theoretical and applied, related to core areas of computer science
2. Demonstrate the ability to work effectively as a team member and/or leader in an ever-changing professional environment
3. Work productively as computer professionals (in traditional careers, graduate school, or academia) by demonstrating effective use of oral and written communication, working competently as a member of a team unit, adhering to ethical standards in the profession.

### **Program Outcomes (POs)**

At the end of the Master of Science (Computer Science) Programme, graduating students/graduates will be able to:

1. Communicate computer science concepts, designs, and solutions effectively and professionally
2. Apply knowledge of computing to produce effective designs and solutions for specific problems
3. Identify, analyse, and synthesize scholarly literature relating to the field of computer science Use software development tools, software systems, and modern computing platforms.
4. Prepare for academic roles through NET/SET/PhD
5. Apply design and development principles in the construction of software systems of varying complexity.

### **Program Specific Outcomes (PSOs)**

1. Demonstrate understanding of the principles and working of the hardware and software aspects of computer systems.
2. Ability to understand the structure and development methodologies of software systems.
3. Possess professional skills and knowledge of software design process. Familiarity and practical competence with a broad range of programming language and open source platforms.
4. Be acquainted with the contemporary issues, latest trends in technological development and thereby innovate new ideas and solutions to existing problems.

**M.Sc-I Semester-I (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**

**Course Code: CC-101**

**Title of Course: Design and Analysis of Algorithms**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course outcomes:**

1. Analyse the asymptotic performance of algorithms.
  2. Demonstrate a familiarity with data structures and algorithms.
  3. Compare algorithms based on time & space complexity.
  4. Employ graphs to model real life problems, when appropriate. Develop algorithms that employ graph computations as key components, and analyse them.
  5. Mapping of data structures like Stack, Queue and Linked List to real life problems.
  6. Be familiar with advanced data structures such as balanced search trees, hash tables, Red-Black trees, Btrees.
  7. Understand Divide & Conquer approach, Greedy algorithm, Backtracking approach for algorithm design.
- 

**UNIT-I**

**[15]**

**Algorithm Analysis:** Introduction to algorithms, analysing and designing algorithms, Growth functions, asymptotic notations, Analysis of recursive algorithms, solving recurrences: Substitution method, recursion tree method, master method. Binary Search, Sorting (Merge Sort, Quick Sort).

**Hashing:** Hashing, Direct address tables, Hash tables, Hash functions, collision resolution techniques.

**Unit-II**

**[15]**

**Data Structures:** Stacks, Queues, Linked list, Trees, Binary tree, Binary search tree, operations on binary search tree, AVL tree, Red-Black Trees, B-trees. **Graphs:** Representations of graph, Traversing Graphs, Breadth-first search, Depth-First Search, topological sort.

**Unit-III**

**[15]**

**Greedy Algorithm:** General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm - Activity selection problem, Elements of Greedy Strategy, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Shortest paths, The Knapsack Problem, Job Scheduling Problem, Huffman code.

**Unit-IV**

**[15]**

**Divide and conquer:** The maximum sub array problem, matrix multiplication, Max-Min problem **Backtracking:** Introduction, N Queen Problem, Subset Sum, Hamiltonian Cycle

**Branch and Bound** – Introduction, 0/1 Knapsack, Travelling Salesman problem,

**Dynamic programming:** Introduction, Tabulation, memoization, Optimal SubstructureProperty in Dynamic Programming

### **References:**

1. Introduction to algorithms, Third Edition. by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, PHI
2. Fundamentals of Computer Algorithms, Second edition. By Ellis Horowitz, Sartaj Sahani, Sanguthevar Rajasekaran, University Press.
3. Data structures and algorithm analysis in C, Second edition. By Mark Allen weiss
4. Fundamental algorithms by Donald E. Knuth, Pearson Education.

**M.Sc-I Semester-I (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**

**Course Code: CC-102**  
**Title of Course: Python Programming**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course Outcomes:**

1. Understand principles of Python
  2. Understand object oriented programming
  3. Demonstrate file handling techniques
  4. Understand how Python can be used for application development
  5. Design Real life problems and think creatively about solution of them
  6. Apply a solution clearly and accurately in a program using python
- 

**Unit-I**

**(15 hrs)**

Installing Python, Simple program using Python, Expressions and Values, Variables and Computer Memory, error detection, Multiple line statements, Designing and using functions, functions provided by Python, Tracing function calls in memory model, omitting return statement. Working with Text: Creating Strings of Characters, Using Special Characters in Strings, Creating a Multiline String , Printing Information, Getting Information from the Keyboard

**Unit-II**

**(15 hrs)**

A Boolean Type, Choosing Statements to Execute, Nested If Statements, Remembering the Results of a Boolean Expression Evaluation, A Modular Approach to Program Organization, Importing Modules , Defining Modules, Testing Code Semi automatically Grouping Functions Using Methods: Modules, Classes, and Methods , Calling Methods the Object-Oriented Way, Exploring String Methods, Underscores.

**Unit-III**

**(15hrs)**

Storing Collections of Data Using Lists: Storing and Accessing Data in Lists, Modifying Lists, Operations on Lists, Slicing Lists, Aliasing, List Methods, Working with a List of Lists. Repeating Code Using Loops: Processing Items in a List, Processing Characters in Strings, Looping Over a Range of Numbers, Processing Lists Using Indices, Nesting Loops in Loops, Looping Until a Condition Is Reached, Repetition Based on User Input, Controlling Loops Using Break and Continue Reading and Writing Files: Kinds of files, Opening a File, Techniques for Reading Files, Files over the Internet, Writing Files, Writing Algorithms that use the File- Reading Techniques, Multiline Records

**Unit-IV**

**(15hrs)**

Storing Data Using Other Collection Types: Storing Data Using Sets, Storing Data Using Tuples, Storing Data Using Dictionaries, Inverting a Dictionary  
Creating Graphical User interface: Building a Basic GUI, Models, Views, and Controllers, Customizing the Visual Style Widgets, Object-Oriented GUIs, Regular expressions Databases: Overview, Creating and Populating, Retrieving Data, Updating and Deleting, Using NULL for Missing Data, Using Joins to Combine Tables, Keys and Constraints.

**References:**

1. Practical Programming: An introduction to Computer Science Using Python, second

edition, Paul Gries, Jennifer Campbell, Jason Montojo, The Pragmatic Bookshelf.

2. Python for Informatics: Exploring Information, Charles Severance
3. Learning Python, Fourth Edition, Mark Lutz, O'Reilly publication
4. Introduction to Python for Computational Science and Engineering (A beginner's guide),  
Hans Fangohr
5. John V Guttag. "Introduction to Computation and Programming Using Python", Prentice  
Hall of India
6. R. Nageswara Rao, "Core Python Programming", Dreamtech



**M.Sc-I Semester-I (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**

**Course Code: CC-103**

**Title of Course: Database Management System**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course Outcomes:**

After successful completion of the course, the student will be able to

1. Define the terminology, features, classifications, and characteristics embodied in database systems.
  2. Demonstrate an understanding of the relational data model.
  3. Transform an information model into a relational database schema and to use a data definition language and/or utilities to implement the schema using a DBMS.
  4. Formulate, using SQL, solutions to a broad range of query and data update problems.
  5. Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.
  6. Use an SQL interface of a multi-user relational DBMS package to create, secure, populate, maintain, and query a database.
  7. Use PL/SQL for handling data in a database as per the user's requirement using programming features.
  8. Define various cursors and its implementation along with procedure and functions.
- 

**UNIT-I**

**[15]**

**Introduction to DBMS:** Database Concept, Characteristics and architecture of DBMS, Database users, 3-tier architecture of DBMS-its advantages over 2-tier, Views of data – Schemas and instances, Data independence. Data models: Conventional data models, NDM & HDM, Physical Data Organization-Hashed files, Indexed files, B-trees

**UNIT-II**

**[15]**

**Database Design and E-R Model:** Entities, Attributes, Relationships, Representation of entities, relationship set, Generalization, aggregation Structure of relational Database and different types of keys, E-R diagrams, E-R design Issues in Relational database design, Functional dependencies. Normalization and Database Recovery systems: Codd's rules, Normalization, Database Recovery System- Failure classification, storage structure, recovery and atomicity, log-based recovery. Role of DBA.

**UNIT-III**

**[15]**

**Introduction to RDBMS:** History, Generations and characteristics, difference between DBMS & RDBMS. Data constraint- primary key, foreign key, unique key, null, not null, default key etc. SQL: Introduction to SQL, Features of SQL, Basic data types, SQL statements/commands, Set operations in SQL, order by and group by clause, like between, in, like, create index, view and join command Nested queries, GRANT and REVOKE, Commit, Rollback, Save point. Join concept: Simple, Equi, non-equi, Self, Outer join. View-Introduction, Create, Update, Drop, Index. SQL functions: MAX, MIN SORT, COUNT, AVERAGE, Numeric, String, Date Functions, Type conversion functions.

**UNIT-IV**

**[15]**

**Introduction to PL /SQL:** Introduction, Difference between SQL AND PL/SQL, Block definition structure and Data types, Block Functions - %Type, %Row Type, Control

statements, Looping statements and sequential statement, Exception handling. Simple PL/SQL blocks.

Cursor management: meaning, types and importance, implicit and explicit cursor management using simple example. Trigger: meaning importance and types of trigger, examples using trigger Procedures-Definition, creating procedures, passing parameters. Function-Definition, syntax and calling methods, passing parameters.

### **Reference Books -**

1. Introduction to database systems C. J. Date Pearsons Education 8th
2. Database system concept Korth, Silberschatz and Sudarshan MGH 5th
3. Fundamentals of Database Systems Elmasri Navathe PearsonEducation5th
4. SQL /PL SQL For Oracle 11G BlackBook Dr.Deshpande WileyDreamtech2012
5. ORACLE PL/SQL Programming Scott Ulman TMH 9th
6. SQL, PL/SQL the programming language of Oracle Ivan Bayross BPB 4th
7. Advance Database Management System  
Chakrabharati/DasguptaWileyDreamtech2011
8. Understanding SQL Martin Gruber BPB 2nd
9. SQL Scott Ulman TMH 4th

**M.Sc-I Semester-I (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**  
**Course Code: CC-104**  
**Title of Course: Cyber Security**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course Outcomes:**

- 1) Realize the need for Cyber Security
  - 2) Understand the need for Security in day to day communications
  - 3) Understand the vulnerabilities in the Network and Computer System
  - 4) Understand the cyber law and Cyber Forensics
- 

**Unit-I**

**(15 hr)**

Introduction to Cyber Security: Overview of Cyber Security, Cyber Threats:- Cyber Warfare-Cyber Crime-Cyber terrorism-Cyber Espionage, Cyber Security Vulnerabilities and Cyber Security Safeguards :Cyber Security Vulnerabilities-Overview, vulnerabilities in software, System administration, Complex Network Architectures, Open Access to Organizational Data, Weak Authentication. Cyber Security Safeguards- Overview, Access control, Audit, Authentication, Biometrics, Cryptography, Deception, Active attacks: Phishing, Sniffing,spoofing, Denial of service attack. Ethical Hacking, Detection Systems, Response, Scanning, Security policy, Threat Management.

**Unit-II**

**(15 hr)**

Securing Web Application, Services and Servers: Introduction, Basic security for HTTP Applications and Services, Basic Security for SOAP Services, Identity Management and Web Services, Authorization Patterns, Security Considerations, Challenges. Intrusion Detection and Prevention: Intrusion, Physical Theft, Abuse of Privileges, Unauthorized Access by Outsider, Malware infection, Intrusion detection and Prevention Techniques, Anti-Malware software, Network based Intrusion detection Systems, Network based Intrusion Prevention Systems, Host based Intrusion prevention Systems, Security Information Management, Network Session Analysis.

**Unit –III**

**(15 hr)**

Cryptography and Network Security: Introduction to Cryptography, Symmetric key Cryptography, Asymmetric key Cryptography, Message Authentication, Digital Signatures, Applications of Cryptography. Overview of Firewalls- Types of Firewalls, VPN Security Protocols: - security at the Application Layer- PGP and S/MIME, Security at Transport Layer- SSL and TLS, Security at Network Layer-IPSec.

**Unit –IV**

**(15 hr)**

Cyberspace and the Law: Introduction, Cyber Security Regulations, Roles of International Law, the state and Private Sector in Cyberspace, Cyber Security Standards. The INDIAN Cyberspace, National Cyber Security Policy 2013.Cyber Forensics: Introduction to Cyber Forensics, Handling Preliminary Investigations, Controlling an Investigation, Conducting disk-based analysis, Investigating Information-hiding, E-mail Security: Encryption for Secure E-mail.

**References:**

1. Preston Gralla, How Personal and Internet Security Work, Que Publications
2. Alfred Basta and Wolf Halton, Computer Security Concepts, Issues and Implementation, Cengage Learning
3. Digital Defense: A Cybersecurity Primer by Joseph Pelton , Indu B. Singh
4. Cryptography and Network Security: Principles and Practice by William Stallings
5. Computer and Information Security Handbook by John R. Vacca .
6. Cyberlaw: The Law of the Internet and Information Technology by Brian Craig .
7. Cyber Warfare: Techniques, Tactics and Tools for Security Practitioners 2nd Edition by Jason Andress (Author), Steve Winterfeld (Author)  
<https://www.studocu.com/en/document/edith-cowan-university/computer-security/lecture-notes/lecture-notes-lectures-1-8-computer-security-notes/709625/view>

**M.Sc-I Semester-I (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**

**Course Code: CCPR-105**

**Title of Course: Python Lab**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course outcomes:**

1. Write, Test and Debug Python Programs
2. Implement Conditionals and Loops for Python Programs
3. Design and implement GUI application and how to handle exceptions and files
4. Use List, Set, Tuples and Dictionaries
5. Read and write data from & to files in Python
6. Make database connectivity in python programming language

---

Lab assignments based on Python Programming course.

**M.Sc-I Semester-I (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**

**Course Code: CCPR-106**

**Title of Course: Database Lab**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course outcomes:**

1. Create, modify and apply operations on table
2. Handling different keys on database
3. Handling different types of SQL queries on database
4. Handling different types of mathematical and statistical functions
5. Handling and write code using PL/SQL
6. Handling cursor in in PL/SQL

---

Lab assignments based on Database Management System course.

**M.Sc-I Semester-I (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**

**Course Code: AEC-107**

**Title of Course: Commutative English-I**

**Syllabus to be implemented from Academic Year 2022-23**

**M.Sc-I Semester-II (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**  
**Course Code: CC-201**  
**Title of Course: Web Technology**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course Outcomes:**

1. To familiarize a student with windows and web-based application
  2. To provide a student with the solid foundation of the syntax and semantics of C# as well as architecture of the .NET framework
  3. Debug and deploy ASP.NET web applications
  4. Discuss the insights of internet programming and implement complete application over the web
  5. To inculcate skills pertaining to data access technology geared to facilitate the development of disconnected systems using .NET platform.
  6. To familiarize the student with the development of windows-based application using C#
  7. To familiarize the student with the development of web-based application using ASP.NET
  8. Handle various toolkit like AJAX
  9. Utilize the concepts of JavaScript
- 

**UNIT -I** **(15 Hrs)**

Introduction: .NET framework and its architecture, CLR, JIT, CTS, Metadata, .NET Revolution, Characteristics of C#, Programming structure of C#, scope of variables, boxing & unboxing, Nullable Data types, Conditional statements, Arrays, Loops, class, inheritance, polymorphism, Exception handling, delegates

**UNIT- II** **(15 Hrs)**

Introduction to C#: Windows form controls, containers, Data Controls, Dialog Controls, sample application development, Deployment of C# application  
ASP.NET: Page Life Cycle, ASP.NET server-side Controls, Data Controls, Server- side Validation Controls, Language Integrated Queries (LINQ), Web services,

**UNIT- III** **(15 Hrs)**

State management: Server side & Client-side state management, Caching in ASP.NET: Page caching, data caching, fragment caching  
AJAX: Introduction to AJAX, Various extenders of AJAX  
ADO.NET: ADO.NET Architecture, ADO.NET namespaces, Working with ADO.NET, Data controls, FormView Control, Working with Repeater, connected architecture, Disconnected Architecture, Three-layer Architecture, Working with Master pages

**UNIT- IV** **(15 Hrs)**

Microsoft SQL Server: Database designing, Tables, Views, Stored Procedures, taking backup of database, creating script of database

Java Script: JavaScript Overview, Data types, variables, scope of variables, casting, data type conversion rules, Expressions and operators. Arrays. Built-in functions, and Built-in objects-String, Date, Math. Three types of dialog boxes-alert, prompt, confirm. Custom Functions.

### **References:**

1. C# 4.0 The Complete Reference by Herbert Schildt
2. Essential C# 4.0” by Mark Michaelis and Eric Lippert
3. “C# in Depth” by Jon Skeet
4. “Head First C#: A Learner’s Guide to Real-World Programming with C#, XAML, and .NET” by Jennifer Greene and Andrew Stellman
5. “Microsoft Visual C# Step by Step” by Sharp John
6. “Let Us C#” by Yashavant P Kanetkar
7. Javascript: The Complete Reference by Thomas Powell & Fritz Schneider
8. JavaScript Pocket Reference, 3rd Edition By David Flanagan
9. Professional JavaScript for Web Developers, 4th Edition by Matt Frisbie

**M.Sc-I Semester-II (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**  
**Course Code: CC-202**  
**Title of Course: Advanced Java**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course Outcomes:**

1. The student will be able to develop distributed business applications, develop web pages using advanced server-side programming through servlets and Java server pages.
  2. Demonstrate approaches for performance and effective coding
  3. Develop Java client/server applications.
  4. Develop distributed applications using RMI
  5. Develop component-based Java software using JavaBeans
  6. Develop server side programs in the form of servlet
  7. Understand the multi-tier architecture of web-based enterprise applications using Enterprise JavaBeans (EJB) ,map Java classes and object associations to relational database tables with Hibernate mapping files
  8. Understand Spring and Spring Boot Framework.
- 

**Unit -1**

**(15 Hrs)**

Overview of Java programming, Java Servlets: Servlet basics, servlet life cycle , Generic and HTTP servlets, The Servlet API, javax.servlet and javax.servlet.http package, session tracking using session and cookies, web deployment descriptor, web.xml. Remote Method Invocation–Introduction, architecture, defining remote objects, creating stubs and skeleton, object serialization, dynamically loaded classes, RMI activation, registering remote objects, marshaled objects.

**Unit-II**

**(15 Hrs)**

Java Database Connectivity.JDBC Overview,Architecture, Types of JDBC Drivers, DriverManager class, database connection statements,ResultSet, transaction, Metadata and Aggregate functions , callable statements, Connection to various back ends. Java Server Pages (JSP): Introduction to JSP tags and directive, Request String, User Sessions, Cookies,Session objects.

**Unit-III**

**(15 Hrs)**

Java Beans: Basics of designing JavaBeans, Java Bean design patterns, creating and using properties, using events to communicate with other components. Enterprise Java Bean: Preparing a Class to be a JavaBean, Creating a JavaBean, JavaBean Properties,Types of beans, Stateful Session bean, Stateless Session bean, Entity bean

**Unit –IV**

**(15 Hrs)**

Spring and Hibernate: Spring API libraries, Designing spring applications. Building a simple application. Introduction to Spring Boot, Features of Spring Boot Spring Boot Architecture, Setup Spring Boot, creating application of spring boot.



**References:-**

1. Java 2 Complete Reference - (Tata McGraw Hill)
2. Java server pages
3. Java 2EE – Ivan Bayross (PHI)
4. Java 2 Black Book –(DreamTech)
5. Orfali, "The essential Distributed Object Survival Guide".
6. Valesky, "Enterprise Java Beans", Addison Wesley.

**M.Sc-I Semester-II (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**

**Course Code: CC-203**

**Title of Course: Android Development with Kotlin**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course outcomes:**

- 1) Comprehend Kotlin language
  - 2) Understand Android Studio Environment and application structure.
  - 3) Demonstrate different layouts, views, activities and intents
  - 4) Design good user interface for the application.
  - 5) Able to store, retrieve and load data
  - 6) Demonstrate programming skills using Kotlin
- 

**Unit 1 – Introduction to Kotlin**

**[15hrs]**

**Kotlin basics:** Introduction to Kotlin, Benefits of using Kotlin, Use Kotlin REPL to practice basic expressions, Control flow statements in Kotlin, Null safety with Kotlin. **Functions:** Creating and calling functions with default and named arguments, Writing concise and compact functions, Passing functions as arguments to other functions, Writing simple lambdas.

**Classes and Objects:** Introduction to object-oriented programming in Kotlin, Classes and objects in Kotlin, Constructors, Visibility modifiers, Subclasses and inheritance, Interfaces, Data classes, Singleton class enums, Pairs, triples and collections in Kotlin, Extensions in Kotlin

**Unit 2 – Introduction to Android**

**[15hrs]**

**Building first Android app:** Installing Android Studio, Creating an Android app project, Deploying the app to an emulator or a device, Building an Android app that contains images and a click handler, Modifying views within the layout of an app, Adding libraries to module gradle file. **Layouts:** Creating layouts in Android Studio using XML and the Layout Editor, Adding interactivity to your app, Working with ConstraintLayout, Data binding basics

**App Navigation:** Creating Fragments, Defining NavHostFragment, navigation graphs, navigational paths, Functionality of Back and Up buttons, Defining the options menu, Creating a navigational drawer, Using the Safe Args plugin and passing of arguments, Starting an external Activity.

**Unit 3 – Android Application Architecture**

**[15hrs]**

**Activity and Fragment Lifecycles:** Understanding Activity and Fragment Lifecycles, Exploring logging options in your app, Using the Android Lifecycle library, Exploring configuration changes

**App Architecture (UI Layer) :** Using the recommended Android App Architecture, Using the Lifecycle, ViewModel, and ViewModelFactory classes, Adding LiveData and

LiveData, observers, Adding Data Binding with ViewModel and LiveData, Adding LiveData, transformations

**App Architecture (Persistence):** Overview of Room Persistence Library, Introduction to coroutines, **Advanced RecyclerView use cases:** Introduction to RecyclerView Fundamentals,

Implementing data binding with RecyclerView, Using GridLayout with RecyclerView, Interacting with RecyclerView items, Adding headers in RecyclerView

## **Unit 4 – Connect to the Internet and App Design [15hrs]**

**Connect to the Internet:** Connecting to a web service with the Retrofit library, Parsing aJSON response with the Moshi library, Using coroutines with Retrofit, Loading and displaying images from the Internet, Filtering data from the Internet.

**Repository pattern and Work Manager:** Adding an offline cache and repository, Implementing Work Manager, Working with background workers and periodic Worker Request

**App UI Design:** Introduction to basic app design, Understanding Styles and Themes, Implementing Material Design, Designing for everyone

### **References:**

1. <https://developer.android.com/kotlin>
2. <https://developer.android.com/kotlin/getting-started-resources>
3. Kotlin for Android Developers: Learn Kotlin the Easy Way While Developing an Android App by , Antonio Leiva (Free ebook )
4. Learn Android Studio 3 with Kotlin: Efficient Android App Development by, Ted Hagos, Apress publisher.
5. [https:// www.shabakeh-mag.com/sites/default/files/files/attachment/1397/04/1530550032.pdf](https://www.shabakeh-mag.com/sites/default/files/files/attachment/1397/04/1530550032.pdf)

**M.Sc-I Semester-II (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**  
**Course Code: CCS-204.1**  
**Title of Course: Software Project Management**

**Syllabus to be implemented from Academic Year 2022-23**

---

Course outcomes:

- 1) To understand Software Project Models and Software Management Concepts.
- 2) To understand the various methods of Cost Estimation.
- 3) To Study about Software Quality Management.
- 4) To Study about Emerging Trends in Software Management.
- 5) To understand Project Evaluation.

---

**Unit-I** **(15)**

**Project Management:** Concept of project Management, Project Organization, Planning a software project, Project management life cycle, Risk management, Identification of Risks, Risk Analysis, Risk Planning and Monitoring.

**Unit-II** **(15)**

**Software Project Estimation:** Concept of Project Estimation, Different methods of software project estimation (COCOMO model, Delphi cost estimation etc.), Function point analysis, Software Project Management Tools and Techniques- PERT & Gantt Charts

**Unit-III** **(15)**

**Software Quality Management and Testing:** Quality Assurance and Standards, Quality Planning, Quality control, Role of testing in Software, development, Testing Procedure, Defect Management. Team Structure, Team Communication, Managing customer expectations, Group Behaviour.

**Unit-IV** **(15)**

**Project Evaluation and Emerging Trends :** Strategic Assessment–Technical Assessment– Cost Benefit Analysis–Cash Flow Forecasting– Cost Benefit Evaluation Technique–Risk Evaluation– Software Effort Estimation. Emerging Trends: people Focused Process Models.

**Reference:**

1. Software Project management By Edwin Bennatan
2. Software Engineering By Roger S. Pressman
3. Software Engineering concepts by Richard Fairley
4. Software Project Management by S.A. Kelkar
5. Software Engineering by IAN Sommerville
6. System Analysis and Design Methods By J.L Whitten , L.D.Bentley and K.C. Dittman
7. Ramesh Gopaldaswamy , “Managing and global Software Projects”, Tata McGraw Hill Tenth Reprint, 2011.

**M.Sc-I Semester-II (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**  
**Course Code: CCS-204.2**  
**Title of Course: Data Science Foundations**

**Syllabus to be implemented from Academic Year 2022-23**

---

Course outcomes:

After successful completion of this course students are able to:

- 1) Understand the concept of Probability and Information Theory
- 2) Compute probabilities of events
- 3) Explain Bayes rule and compute probabilities using Bayes rule
- 4) To formulate and solve classification problem
- 5) To investigate clustering techniques for particular data
- 6) To mine frequent pattern in a dataset
- 7) To get acquainted with natural language processing techniques

---

**Unit – I**

**(15hrs)**

Probability and Information Theory: Introduction to Probability, Random Variables, probability Distributions, Marginal Probability, Conditional Probability, The Chain Rule of Conditional Probabilities, Independence and Conditional Independence, Expectation, Variance and Covariance, Common Probability Distributions, Useful Properties of Common Functions, Bayes' Rule, Information theory, Structured Probabilistic Models

**Unit – II**

**(15hrs)**

Data understanding and data cleaning, concept of supervised and unsupervised learning. Classification techniques: Classification: Preliminaries, general approach to solve classification problem, Decision tree induction, Rule-based classifier, Nearest-Neighbor classifier, Bayesian Classifiers, Support Vector Machine.

**Unit – III**

**(15hrs)**

Cluster analysis: Introduction, Types of Clustering, Types of Clusters. K-means algorithm, Agglomerative Hierarchical Clustering, DBSCAN, Prototype based clustering and Density based clustering, introduction to Web Mining. Association analysis: Problem definition, Frequent Itemset Generation, Apriori Principle, apriori algorithm, Maximal Frequent itemset, closed frequent itemset. FP-growth algorithm

**Unit – IV**

**(15hrs)**

Introduction to Natural Language Processing (NLP), Data Preparation: punctuation removal, stop-words removal, numeric value removal, frequent words removal, rare words removal, spelling correction, tokenization, stemming, lemmatization. Feature Engineering: count vectors as features, Term Frequency-Inverse Document Frequency (TF-IDF), TF-IDF vectors as features, word level TF-IDF, N-Gram level, TF-IDF, Character level TF-IDF, Inverse Document Frequency, word embedding as features, Text/NLP based features, Topic Models as features, word2vec.

**References:**

- 1) Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc."

- 2) Data Mining concepts and techniques --- Jiawei Han and Micheline Kamber , Elsevier
- 3) Data Mining: Introductory and Advanced Topics - Margaret H. Dunham, Pearson education
- 4) Data Mining: Practical Machine Learning Tools and Techniques, Ian H. Witten, Eibe Frank
- 5) Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). Cambridge: MIT press.
- 6) Introduction to Data Mining – Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson education.
- 7) Kumar, E. (2013). *Natural language processing*. IK International Pvt Ltd.
- 8) Ross, S. M. (2014). *Introduction to probability models*. Academic press.

**M.Sc-I Semester-II (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**

**Course Code: CCS-204.3**

**Title of Course: Application Security Analyst**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course outcomes:**

- 1) Understand application development life cycle
  - 2) Identify vulnerabilities, entry points, error code etc.
  - 3) Get acquainted with password cracking techniques and prevention techniques.
  - 4) Understand authentication and authorization vulnerabilities.
  - 5) Demonstrate use of SQL injection.
  - 6) Investigate Session & browser manipulations
- 

**Unit-I**

**(15hrs)**

**Introduction:** standard Systems Development Lifecycle (SDLC) practices and process, enterprise information technology (IT) architecture, key vulnerabilities to applications , Risk management frameworks and processes, Site Mapping & Web Crawling, Server & Application Fingerprinting, Identifying the entry points, File extensions handling, Page enumeration and brute forcing, **Comments in code** – view source, Looking for leftovers and backup files, Admin interfaces, Robots.txt, Analysis of error code

**Unit-II**

**(15hrs)**

**Basic security concepts:** hardware and software vulnerabilities

**Malware:** Types of Malware, Virus, Trojan, Key logger, File Binder, Testing FUD and Countermeasure.

**Password Cracking and Prevention:** Introduction, Password Cracking Techniques, Dictionary Based Attack, Brute force Attack, Cracking common files password, Cracking web based password, Password reset flaws, Password change flaws, Cracking Wifi password, Countermeasures for users, Countermeasures for System Administrators.

**Unit-III**

**(15hrs)**

**Authentication & Authorization vulnerabilities:** Authentication concepts, scenarios, User enumeration, Direct page requests, Parameter modification, Lack of SSL at login pages , Bypassing weak CAPTCHA mechanisms, Login without SSL, Authorization: RBAC, Authorization bypassing, Parameter tampering, Forceful browsing, Rendering based Authorization, Client side validation attacks, Insecure direct object reference.

**File handling:** Insecure file handling, Path traversal, Canonicalization, Uploaded files backdoors, Insecure file extension handling, Directory listing, File size, File type, Malware upload

**Unit-IV**

**(15hrs)**

**Input vulnerabilities:** SQL injection, Common implementation mistakes - authentication bypassing using SQL Injection, Cross Site Scripting, Reflected VS. Stored XSS

Command injection, Special chars - '& < >', empty

**Session & browser manipulation:** Session & browser manipulation attacks, Session management techniques, Cookie based session management, , cookie properties, Cookies - secrets in cookies, tampering, Exposed session variables, Missing Attributes – httpOnly, secure Session validity after logoff, Long session timeout, Session keep alive – enable/disable, Session id rotation, Cross Site Request Forgery (CSRF), Open redirect , **IT Auditing**–Definition, Objectives and strategies

## **References:**

1. Bryan Sullivan and Vincent Liu, Web Application Security, A Beginner's Guide, Mc Graw Hill
2. Michael Cross, Developer's Guide to Web Application Security, Syngress Publishers
3. Justin Clarke , SQL Injection Attacks and Defense, Elsevier
4. Nigel Chapman, Authentication and Authorization on the Web, Publisher Macavon Media, 2012
5. <https://owasp.org/>



**M.Sc-I Semester-II (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**  
**Course Code: CCS-204.4**  
**Title of Course: Cloud Computing**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course outcomes:**

- 1) Introduce the broad perceptive of cloud architecture and model
- 2) Apply different cloud programming model as per need.
- 3) Explore some important cloud computing driven commercial systems such as Google Apps, Microsoft Azure and Amazon Web Services and other businesses cloud applications
- 4) To learn how to use Cloud Services.
- 5) To implement Task Scheduling algorithms.
- 6) To build Private Cloud.

---

**Unit – I: (15hrs)**

**Fundamentals of Cloud Computing:** Overview of Cloud Computing, Evolution of Cloud Computing, Types of Clouds, Key Characteristics of Cloud Computing, Intranets and Clouds. Benefits and challenges of cloud computing, Usage scenarios and Applications, Regulatory issues, major players in cloud computing.

**Unit – II (15hrs)**

**Cloud Models & Services :** Cloud Models – Benefits of Cloud Models, Public, Private, Hybrid, and Community Clouds, Types of Clouds Services: SaaS, PaaS, IaaS, DaaS, MaaS, CaaS. Service Providers: Google, App Engine, Microsoft Azure, Amazon EC2, IBM, Salesforce; Introduction to MapReduce, GFS, HDFS, Hadoop Framework.

**Unit – III (15hrs)**

**Essentials & Collaborating with Cloud:** Hardware and Infrastructure – Clients, Security, Network, Services; Accessing Cloud – Platforms, Web Applications, Web APIs, Web Browsers; Cloud Storage – Overview, Cloud Storage Providers; Standards – Application, Client, Infrastructure, Service; Centralizing Email Communications, Collaborating on Calendars, Schedules & Task Management, Event Management, Project Management and Contact Management.

**Unit-IV (15hrs)**

**Virtualization & Security for Clouds:** Need for Virtualization – Pros and Cons of Virtualization, Types of Virtualization, System VM, Process VM, Virtual Machine Monitor – Virtual Machine Properties, HLL VM, Hypervisor – VMWare, Virtual Box, Hyper-V; Case Studies on Cloud Data Centres. Security in Clouds – Cloud security challenges, SaaS as Service Security; Common Standards – Open Cloud Consortium, Distributed Management Task Force, Standards for Application Developers; Standards for Messaging – Standards for Security, End User access to cloud computing, mobile internet devices and the cloud.

**Reference Books:**

1. Bloor R., Kanfman M., Halper F. Judith Hurwitz “Cloud Computing for Dummies” (Wiley India Edition) 2010, ISBN 978-0-470-48470-8.
2. Barrie Sosinsky, “Cloud Computing Bible, “ Wiley India Pvt. Ltd. 2012
3. George Reese, “Cloud Application Architectures”, Shroff/O’ Reilly, 2009.

4. John W. Rittinghouse & James F. Ransome, "Cloud Computing: Implementation, Management and Security", CRC Press, 1st Edition, 2009, ISBN 978-1439806807.
5. Antohy T Velte, Toby J. Velte, Robert Elsenpeter, Cloud Computing: "A Practical Approach", McGraw Hill, 2009. ISBN 978-0-07-068351-8
6. Michael Miller, Cloud Computing: "Web-Based Applications That Change the Way You Work and Collaborate Online", Que Publishing, August 2008. ISBN 978-0-7897-3803-5
7. George Reese, "Cloud Application Architecture", O'Reilly and Associates.

**M.Sc-I Semester-II (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**  
**Course Code: CCPR-205**  
**Title of Course: Web Technology Lab and Advanced Java Lab**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course outcomes:**

- 1) Debug and deploy ASP.NET web applications
- 2) Discuss the insights of internet programming and implement complete application over the web
- 3) Use the features of Dot Net Framework along with the features of C#
- 4) Build and host web applications using ASP.NET
- 5) Develop and deploy Windows applications
- 6) Handle data by using ADO.NET architecture
- 7) Create database-driven ASP.NET web applications and web services
- 8) Handle various toolkit like AJAX
- 9) Utilize the concepts of JavaScript
- 10) Develop and deploy a website using HTML
- 11) Define & explain applet Life cycle
- 12) Differentiate local and remote applet
- 13) Write the code for a simple Java applet
- 14) Explain applet tag and its parameter
- 15) Use the methods of the Applet and Component classes required for a basic applet
- 16) Describe the classes in the AWT package that relate to the Applet class
- 17) Describe the AWT graphics explain controls and how to apply them in the container
- 18) Develop simple programs using Event class and Event Listener Interface
- 19) Develop a program for steps to connect a database
- 20) Describe the Basics of JDBC
- 21) Explain the different Types of JDBC drivers & their advantages and Disadvantages
- 22) Develop program to use JDBC to query a database and modify
- 23) Describe life cycle of servlet
- 24) Develop program using javax.servlet package
- 25) Explain JSP Architecture and its Life cycle
- 26) Develop simple program

**M.Sc-I Semester-II (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**  
**Course Code: CCPR-206**  
**Title of Course: Project**

**Syllabus to be implemented from Academic Year 2022-23**

---

**Course outcomes:**

- 6) Gain skills as they apply knowledge effectively in diverse contexts.
- 7) Analyse and model requirements and constraints for the purpose of designing and implementing software artefacts and IT systems
- 8) Design and implement software solutions that accommodate specified requirements and

constraints, based on analysis or modelling or requirements specification

- 9) Present a clear, coherent and independent exposition of software applications, alternative IT solutions, and decision recommendations to both IT and non-IT personnel via technical reports of professional standard and technical presentations.
- 10) Team work: Work effectively in different roles, to form, manage, and successfully produce outcomes from teams, whose members may have diverse cultural backgrounds and life circumstances, and differing levels of technical expertise.

**A software module development should be carried out as part of Project work.**

**M.Sc-I Semester-II (Computer Science)**  
**Choice Based Credit System with Multiple Entry and Multiple Exit Option**  
**(NEP-2020)**  
**Course Code: SEC-207**  
**Title of Course:**

**Syllabus to be implemented from Academic Year 2022-23**