HSNC UNIVERSITY, MUMBAI

Board of Studies in Faculty of Science & Technology In the subject of Computer Science, KC College

1. **Name of Chairperson : -**

a. **Ms. Geeta N. Brijwani,** Assistant Professor, Dept. of Comp. Sci., KC College, HSNC University, Mumbai.

2. **Two to five teachers each having minimum five years teaching experience amongst the full time teachers of the Departments, in the relevant subject.**

a. **Mrs. Shalini A. Maheshgauri,** Assistant Professor, Dept. of Comp. Sci., KC College, HSNC University, Mumbai.

b. **Mrs. Beenarani S. Karutharan,** Assistant Professor, Dept. of Comp. Sci., KC College, HSNC University, Mumbai.

c. **Mr. Naveen Pahuja,** Assistant Professor, Dept. of Comp. Sci., KC College, HSNC University, Mumbai.

d. **Mr. Vipul Saluja,** Assistant Professor, Dept. of Comp. Sci., R.D. National College, Bandra(West).

3. **One Professor / Associate Professor from other Universities or professor / Associate Professor from colleges managed by Parent Body; nominated by Parent Body;-**

a. **Dr. Sushil Kulkarni,** Associate Professor, Head, Dept. of Mathematics, Jai Hind College Autonomous, Mumbai.

4. **Four external experts from Industry / Research / eminent scholar in the field relevant to the subject nominated by the Parent Body;**

a. **Mr. Shreekant Shiralkar,** Head - India Solution Center for SAP at Tata Consultancy, Mumbai.

b. **Mr. Harish Chandar,** Director, India Tech International Pvt. Ltd., Mumbai.

c. **Dr. Subodh Deolekar,** Lead Research Engineer at REDx We School & Assistant Professor at Research and Business Analytics, Prin. L. N. Welingkar Institute of Management Development & Research.

d. **Mr. Roy Thomas,** Head, Dept. of IT, Xavier's College – Autonomous, Mumbai.

5. Top rankers of the Final Year Graduate and Final Year Post Graduate examination of previous year of the concerned subject as invitee members for discussions on framing or revision of syllabus of that subject or group of subjects for one year.

a. **Mr. Maunash A. Jani**, Software Developer, Genius Lynx, Mumbai.

b. **Mr. Ajit Vishwakarma,** Corporate Master Trainer, Managing Director, Vinayavish LLP, Mumbai.

Invitee BOS Members

a. **Dr. Jyotshna Dongardive,** Assistant Professor, University Department of Computer Science, Kalina, Santacruz (East).

b. **Mr. Rajesh Maurya,** Assistant Professor, Dept. of IT, SVKM's Usha Pravin Gandhi College, Vile Parle (West).

c. **Dr. Madhavi Vaidya,** Assistant Professor, Dept. of Comp. Sci., Vivekanand Education Society, Chembur.

d. **Dr. Girish Tere,** Assistant Professor, Dept. of Comp. Sci., Thakur College, Kandivili.

e. **Dr. Amol Joglekar,** Assistant Professor, Dept. of Comp. Sci., Mithibai College, Vile Parle.

Part –I

Outline of Choice Based Credit System as outlined by University Grants Commission:

R. **: The Definitions of The Key Terms Used in The Choice Based Credit System And Grading System Introduced From TheAcademicYear2020-2021AreAs Under:**

1. **Core Course:** A course, which should compulsorily be studied by a candidate as a core requirement is termed as a Core course.

2. **Elective Course:** Generally, a course which can be chosen from a pool of courses and which may be very specific or specialized or advanced or supportive to the discipline/subject of study or which provides an extended scope or which enables an exposure to some other discipline/subject/domain or nurtures the candidate's proficiency/skill is called an Elective Course.

   2.1 **Discipline Specific Elective (DSE) Course**: Elective courses may be offered by the main discipline/subject of study is referred to as Discipline Specific Elective. The University/Institute may also offer discipline related Elective courses of interdisciplinary nature (to be offered by main discipline/subject of study).

   2.2 **Dissertation/Project:** An elective course designed to acquire special/advanced knowledge, such as supplement study/support study to a project work, and a candidate studies such a course on his own with an advisory support by a teacher/faculty member is called dissertation/project. A Project/Dissertation work would be of 6 credits. A Project/Dissertation work may be given in lieu of a discipline specific elective paper.

   2.3 **Generic Elective (GE) Course:** An elective course chosen generally from an unrelated discipline/subject, with an intention to seek exposure is called a Generic Elective. P.S.: A core course offered in a discipline/subject may be treated as an elective by other discipline/subject and vice versa and such electives may also be referred to as Generic Elective.

3. **Choice Base Credit System:** CBCS allows students to choose inter- disciplinary, intra-disciplinary courses, skill-oriented papers (even from other disciplines according to their learning needs, interests and aptitude) and more flexibility for students.
4. **Honors Program:** To enhance employability and entrepreneurship abilities among the learners, through aligning Inter Disciplinary / Intra Disciplinary courses with Degree Program. Honours Program will have 40 additional credits to be undertaken by the learner across three years essentially in Inter / Intra Disciplinary course.

A learner who joins Regular Undergraduate Program will have to opt for Honours Program in the first year of the Program. However, the credits for honours, though divided across three years can be completed within three years to become eligible for award of honours Degree.

5. **Program:** A Program is a set of course that are linked together in an academically meaningful way and generally ends with the award of a Degree Certificate depending on the level of knowledge attained and the total duration of study, B.Sc. Programs.

6. **Course:** A 'course' is essentially a constituent of a 'program' and may be conceived of as a composite of several learning topics taken from a certain knowledge domain, at a certain level. All the learning topics included in a course must necessarily have academic coherence, i.e. there

must be a common thread linking the various components of a course. A number of linked courses considered together are in practice, a 'program'.

7. **Bridge Course:** Bridge course is visualized as Pre semester preparation by the learner before commencement of regular lectures. For each semester the topics, whose knowledge is considered as essential for effective and seamless learning of topics of the Semester, will be specified. The Bridge Course can be conducted in online mode. The Online content can be created for the Bridge Course Topics.

8. **Module and Unit:** A course which is generally an independent entity having its own separate identity, is also often referred to as a 'Module' in today's parlance, especially when we refer to a 'modular curricular structure'. A module may be studied in conjunction with other learning modules or studied independently. A topic within a course is treated as a Unit. Each course should have exactly 3 Units.

9. **Self-Learning: 20% of the topics will be marked for Self-Learning.** Topics for Self-Learning are to be learned independently by the student, in a time- bound manner, using online and offline resources including online lectures, videos, library, discussion forums, fieldwork, internships etc.

Evaluative sessions (physical/online), equivalent to the credit allocation of the Self Learning topics, shall be conducted, preferably, every week for each course. Learners are to be evaluated real time during evaluative sessions. The purpose of evaluative sessions is to assess the level of the students' learning achieved in the topics are marked for Self-Learning.

The teacher's role in these evaluative sessions will be that of a Moderator and Mentor, who will guide and navigate the discussions in the sessions, and offer concluding remarks, with proper reasoning on the aspects which may have been missed by the students, in the course of the Self-Learning process.

The modes to evaluate self-learning can be a combination of the various methods such as written reports, handouts with gaps and MCQs, objective tests, case studies and Peer learning. Groups can be formed to present self- learning topics to peer groups, followed by Question-and-Answer sessions and open discussion. The marking scheme for Self-Learning will be defined under Examination and Teaching.

The topics stipulated for self-learning can be increased or reduced as per the recommendations of the Board of Studies and Academic Council from time to time. All decisions regarding evaluation need to be taken and communicated to the stakeholders preferably before the commencement of a semester. Some exceptions may be made in exigencies, like the current situation arising from the lockdown, but such adhoc decisions are to be kept to the minimum possible.

10. **Credit Point:** Credit Point refers to the 'Workload' of a learner and is an index of the number of learning hours deemed for a certain segment of learning. These learning hours may include a variety of learning activities like reading, reflecting, discussing, attending lectures / counseling sessions, watching especially prepared videos, writing assignments, preparing for examinations, etc. Credits assigned for a single course always pay attention to how many hours it would take for a learner to complete a single course successfully. A single course should have, by and large a course may be assigned anywhere between 2 to 8 credit points wherein 1 credit is construed as corresponding to approximately 30 to 40 learning hours.

11. **Credit Completion and Credit Accumulation:** Credit completion or Credit acquisition shall be considered to take place after the learner has successfully cleared all the evaluation criteria with respect to a single course. Thus, a learner who successfully completes a 4 CP (Credit Point) course may be considered to have collected or acquired 4 credits. learner level of performance above the minimum prescribed level (viz. grades / marks obtained) has no bearing on the number of credits collected or acquired. A learner keeps on adding more and more credits as he completes successfully more and more courses. Thus, the learner 'accumulates' course wise credits.

12. **Credit Bank:** A Credit Bank in simple terms refers to stored and dynamically updated information regarding the number of Credits obtained by any given learner along with details regarding the course/s for which Credit has been given, the course-level, nature, etc. In addition, all the information regarding the number of Credits transferred to different programs or credit exemptions given may also be stored with the individual's history.

13. **Credit Transfer:** (performance transfer) When a learner successfully completes a program, he/she is allowed to transfer his/her past performance to another academic program having some common courses and Performance transfer is said to have taken place.

14. **Course Exemption:** Occasionally, when two academic programs offered by a single university or by more than one university, may have some common or equivalent course-content, the learner who has already completed one of these academic programs is allowed to skip these 'equivalent' courses while registering for the new program. The Learner is 'exempted' from 'relearning' the common or equivalent content area and from re-appearing for the concerned examinations. It is thus taken for granted that the learner has already collected in the past the credits corresponding to the exempted courses.

Part-II

O*** The fees for transfer of credits or performance will be based on number of credits that a learner has to complete for award of the degree.**

The Scheme of Teaching and Examination:

The performance of the learners shall be evaluated in two components: Internal Assessment with 40% marks by way of continuous evaluation and by Semester End Examination with 60% marks by conducting the theory examination.

INTERNAL ASSESSMENT: - It is defined as the assessment of the learners on the basis of continuous evaluation as envisaged in the credit-based system by way of participation of learners in various academic and correlated activities in the given semester of the programme.

A). Internal Assessment– 40% 40 marks

Practical's (internal Components of the Practical Course

1. For Theory Courses

Sr. No.	Particulars	Marks
1	ONE class test/online examination to be conducted in the given semester	15 Marks
2	One assignment based on curriculum (to be assessed by the teacher Concerned	10 Marks
3	Self-Learning Evaluation	10 Marks

4	Active participation in routine class instructional deliveries	05 Marks

2. For Courses with Practicals

Each practical course can be conducted out of 50 marks with 20 marks for internal and 30 marks for external

Practical's (Internal component of the Practical Course)

Sr. No	Evaluation type	Marks
1	Two Best Practicals /Assignments/Presentation /Preparation of models/ Exhibits **Or** One Assignment/ project with class presentation to be assessed by teacher concerned	10
2	Journal	05
3	Viva	05

The semester end examination (external component) of 60 % for each course will be as follows:

i) **Duration – 2 Hours** ii) **Theory Question Paper**

Pattern: -

1. There shall be four questions each of 15 marks. On each unit there will be one question and the fourth one will be based on entire syllabus.
2. All questions shall be compulsory with internal choice within the questions. (Each question will be of 20 to 23 marks with options.)
3. Question may be subdivided into sub-questions a, b, c… and the allocation of marks depend on the weightage of the topic.

The marks will be given for all examinations and they will be converted into grade (quality) points. The semester-end, final grade sheets and transcripts will have only credits, grades, grade points, SGPA and CGPA.

3. **Project and Assignment:**

Project or Assignment, which can in the following forms

- Case Studies
- Videos
- Blogs
- Research paper (Presented in Seminar/Conference)
- Field Visit Report
- Presentations related to the subject (Moot Court, Youth Parliament, etc.)
- Internships (Exposition of theory into practice)
- Open Book Test
- any other innovative methods adopted with the prior approval of Director Board of Examination and Evaluation.

4. **Self-Learning Evaluation**

- **20% OF THE TOPICS OF CURRICULUM ARE LEARNED BY THE STUDENT THROUGH SELF LEARNING USING ONLINE / OFFLINE ACADEMIC RESOURSE SPECIFIED IN THE CURRICULUM.**

- **HENCE 20% OF THE LECTURES SHALL BE ALLOCATED FOR EVALUATION OF STUDENTS ON SELF LEARNING TOPICS**

- The identified topics in the syllabus shall be learnt independently by the students in a time bound manner preferably from online resources. Evaluative sessions shall be conducted by the teachers and will carry 10 Marks.

- CLUB The self-learning topics into 3-4 GROUPS OF TOPICS ONLY FOR EVALUATION.

- **PRESCRIBE TIME DURATION (IN DAYS) FOR COMPLETION OF EACH GROUP OF TOPIC AND EARMARK SELF LEARNING EVALUATION LECTURES IN THE TIMETABLE. HENCE EACH GROUP OF TOPIC CAN BE ASSIGNED 3 REGULAR LECTURES FOR THIS EVALUATION FOR ENTIRE CLASS**

  3 Sub Topics
  Each evaluative session shall carry 3 Marks (3 x 3 Units = 9 Marks). Students who participate in all evaluative sessions shall be awarded 1 additional Mark.
  4 Sub Topics
  Each evaluative session shall carry 2.5 Marks (2.5 x 4 Units = 10 Marks)

- **EVALUATION OF SELF LEARNING TOPICS CAN COMMENCE IN REGULAR LECTURES ASSIGNED FOR SELF LEARNING EVALUATION IN THE TIMETABLE**

  3 Evaluative sessions
  Each evaluative session shall carry 3 Marks (3 x 3 = 9 Marks). Students who participate in all evaluative sessions shall be awarded 1 additional Mark
  4 **Evaluative sessions**
  Each evaluative session shall carry 2.5 Marks (2.5 x 4 = 10 Marks).

  <u>Methods for Evaluation of Self-learning topics:</u>

- Seminars/presentation (PPT or poster), followed by Q&A − Objective questions /Quiz / Framing of MCQ questions.
- Debates
- Group discussion
- You-Tube videos (Marks shall be based on the quality and viewership)
- Improvisation of videos
- Role Play followed by question-answers
-

TEACHERS CAN FRAME OTHER METHODS OF EVALUATION ALSO PROVIDED THAT THE METHOD, DULY APPROVED BY THE COLLEGE EXAMINATION COMMITTEE, IS NOTIFIED TO THE STUDENTS AT LEAST 7 DAYS BEFORE THE COMMENCEMENT OF THE EVALUATION SESSION AND IS FORWARDED FOR INFORMATION AND NECESSARY ACTION AT LEAT 3 DAYS BEFORE THE COMMENCEMENT OF THE EVALUATION SESSION

- Viva Voce
- Any other innovative method

SEMESTER END EXAMINATION: - It is defined as the examination of the learners on the basis of performance in the semester end theory / written examinations.

B. Semester End Examination-60 % ***60 Marks***

1) Duration – These examinations shall be of 2 Hours duration.
2) Question Paper Pattern: -
   i. There shall be four questions each of 15 marks.
   ii. All questions shall be compulsory with internal choice within the questions.
   iii. Question may be sub-divided into sub-questions a, b, c, d & e only and the allocation of marks depends on the weightage of the topic.

THE MARKS OF THE INTERNAL ASSESSMENT SHOULD NOT BE DISCLOSED TO THE STUDENTS TILL THE RESULTS OF THE CORRESPONDING SEMESTER IS DECLARED.

HSNC University Mumbai

(2021-2022)

Ordinances and Regulations

With Respect to

Choice Based Credit System (CBCS)

For the Programmes Under

The Faculty of Science and Technology

For the Course

Computer Science

Curriculum – Second Year Undergraduate Programmes Semester-III and Semester -IV

2021-2022

Section D Computer Science Part 1- Preamble

Information and Communication Technology (ICT) has today become integral part of all industry domains as well as fields of academics and research. The industry requirements and technologies have been steadily and rapidly advancing. Organizations are increasingly opting for open source systems. The students too these days are thinking beyond career in the industry and aiming for research opportunities. B.Sc. Computer Science programme is designed to cover all aspects of computer knowledge required to prepare students for successful careers in the software industry.

1. Course Objective

i. Inculcating high knowledge levels of software development and programming languages.

ii. Provide students with the tools that will allow them to design and implement software solutions to problems.

iii. To have hands on experience in developing a software project by using various Software Engineering principles and methods in each of the phases of software development.

iv. To spark the ambition towards their own constant and ongoing professional development.

The programme of BSc. Computer Science will enable students to be placed in different fields such as:

- Website Development / Mobile Application Development
- Database Administration
- Automated IOT Systems
- Game designing
- Software Testing
- Artificial Intelligence
- Data Science
- Linux Server Administration
- Information Security
- Ethical hacking

2. Process adopted for curriculum designing

- The final programme was outlined after frequent discussions, meetings, brainstorming sessions and electronic interactions with academic, alumni and industry partners.
-

3. Salient features Syllabus made more relevant

- Restructures the existing curriculum.
- Introduction of new subjects/concepts in courses to furnish students with the skills needed to contribute in an ever evolving IT field.
- Identify and nurture research temper among students.
- Improvement in the employability skills.
- Culture of Innovation and Critical Thinking.
- Relevant to the contemporary & emerging needs of employers.

4. Learning Outcomes

- Students will attain techniques, skills, and tools necessary for computing practice and development.
- Students will be able to apply computing theory and programming principles to practical software design and development.
-

5. Input from stakeholders with relevant information

- After discussion with stakeholders, certain changes were brought in few topics in the current subjects while a few new subjects are introduced such as “PHP with E- Commerce” which introduces PHP as a general-purpose scripting language, Laravel – free & open-source PHP web framework and WooCommerce. Another subject “Introduction to Research Methodology” is introduced which develops skills in data analysis, critical thinking and technical writing. Few subject(s) such as "Software Engineering", "Software Testing and Quality Assurance" are shifted from higher semesters to Semester- III and Semester - IV so as to introduce new subject(s) in the final year.

Part 2 - The Scheme of Teaching and Examination is as under:
Semester – III Summary

Sr. No.	**Choice Based Credit System**			**Subject Code**	**Remarks**
1	**Core Course (Computer Science)**			US-SCS-301 US-SCS-302 US-SCS-303 US-SCS-304	
2	**Elective Course**	**Discipline Specific Elective (DSE) Course**			
		2.1	**Interdisciplinary Specific Elective (IDSE) Course**		
		2.2	**Dissertation/Project**		
		2.3	**Generic Elective (GE) Course**	US-SCS-305	
3	**Ability Enhancement Courses (AEC)**				
4	**Skill Enhancement Courses (SEC)**			US-SCS-306 US-SCS-307	

Second Year Semester III Internal and External Detailed Evaluation Scheme

Sr. No.	**Subject Code**	**Subject Title**	**Periods Per Week**					**Credits**	**Internals**				**Total Marks**
			Units	**S.L.**	**L**	**T**	**P**		**SLE**	**CT + AT = 15 + 5**	**PA**	**SEE**	
1	US-SCS-301	Theory of Computation	3	20% *	3	0	-	2	10	20	10	60	100
2	US-SCS-302	Core Java	3	20% *	3	0	-	2	10	20	10	60	100
3	US-SCS-303	Operating Systems	3	20% *	3	0	-	2	10	20	10	60	100
4	US-SCS-304	Software Engineering	3	20% *	3	0	-	2	10	20	10	60	100
5	US-SCS-305	Physical Computing and IoT Programming	3	20% *	3	0	-	2	10	20	10	60	100
6	US-SCS-306	PHP with E-Commerce	3	20%	3	0	-	2	10	20	10	60	100
7	US-SCS-307	Android Developer Fundamentals	3	20%	3	0	-	2	10	20	10	60	100
8	US-SCS-3P1	Practical of US-SCS-302 + US-SCS-303 + US-SCS-304	-	-	-	-	9	3				150 (90+ 60)	150
9	US-SCS-3P2	Practical of US-SCS-305 + US-SCS-306 + US-SCS-307	-	-	-	-	9	3				150 (90+ 60)	150
	Total Lectures/ Credits							**20**	**Total Marks**				**1000**

***One to two lectures to be taken for CONTINUOUS self –learning Evaluation.**

Second Year Semester III - Units – Topics – Teaching Hours

Sr. No.	Subject Code & Title	Subject Unit Title		Hours/ Lectures	Total No. of hours/ lectures	Credit	Total Marks
1	US-SCS-301 Theory of Computation	1	Automata Theory, Formal Languages	15	45L	2	100 (60+40)
		2	Regular Sets and Regular Grammar, Context Free Languages, Pushdown Automata	15			
		3	Linear Bound Automata, Turing Machines, Undecidability	15			
2	US-SCS-302 Core Java	1	The Java Language, OOPS, String Manipulations, Packages	15	45L	2	100 (60+40)
		2	Exception Handling, Multithreading, I/O Streams, Networking	15			
		3	Wrapper Classes, Collection Framework, Inner Classes, AWT	15			
3	US-SCS-303 Operating Systems	1	Introduction and Operating-Systems Structures, Operating-System Structures, Processes, Threads	15	45L	2	100 (60+40)
		2	Process Synchronization, CPU Scheduling, Deadlocks	15			
		3	Main Memory, Virtual Memory, Mass-Storage Structure, File-System Interface, File-System Implementation	15			
4	US-SCS-304 Software Engineering	1	Introduction, Requirement Analysis & System Modeling	15	45L	2	100 (60+40)
		2	System Design, Software Measurement and Metrics	15			
		3	Software Project Management, Risk Management	15			
5	US-SCS-305 Physical Computing and IoT Programming	1	System on Chip, SoC Products, ARM8 Architecture, Introduction to Raspberry Pi, Raspberry Pi Boot	15	45L	2	100 (60+40)
		2	Raspberry Pi and Linux, Programming Interfaces, Raspberry Pi Interfaces, Useful Implementations	15			
		3	Introduction to IoT, IoT Security, IoT Service as a Platform, IoT Security and Interoperability.	15			
6	US-SCS-306 PHP with E-Commerce	1	PHP, PHP OOP	15	45L	2	100 (60+40)
		2	PHP & MySQL Database using MySQL and PHP Data Objects (PDO), Laravel	15			

		3	Introduction to Ecommerce, EDI, WooCommerce/Magento	15			
7	US-SCS-307 Android Developer Fundamentals	1	Introduction to Android, Activities & Intents, Testing, debugging, and using support libraries	15	45L	2	100 (60+40)
		2	User Experience- User Interaction, Delightful User Experience, Testing your UI	15			
		3	Working in the background Tasks, Alarms and Schedulers, Saving User Data- Preferences & Settings, Storing Data with Room	15			
8	US-SCS-3P1	1	Practical based on US-SCS-302	3	45x3= 135 lectures per batch	3	150 (90+60)
		2	Practical based on US-SCS-303	3			
		3	Practical based on US-SCS-304	3			
9	US-SCS-3P2	1	Practical based on US-SCS-305	3	45x3= 135 lectures per batch	3	150 (90+60)
		2	Practical based on US-SCS-306	3			
		3	Practical based on US-SCS-307	3			
			TOTAL			**20**	**1000**

- **Lecture Duration – 48 Minutes**
- **One Credit =15 hours**

L: Lecture: Tutorials P: Practical Ct-Core Theory, Cp-Core Practical, SLE- Self learning evaluation CT-Commutative Test, SEE- Semester End Examination, PA-Project Assessment, AT- Attendance

Part - 3 Detailed Scheme Theory

Curriculum Topics along with Self-Learning topics to be covered, through self-learning mode along with the respective Unit. Evaluation of self-learning topics to be undertaken before the concluding lecture instructions of the respective UNIT.

Course Code: US-SCS-301 Theory of Computation

Unit No.	**Content**	**No. of Lectures**
I	**1.1 Automata Theory:** Defining Automaton, Finite Automaton, Transitions and Its properties, Acceptability by Finite Automaton, Nondeterministic Finite State Machines, DFA and NDFA equivalence, Mealy and Moore Machines, Minimizing Automata. **1.2 Formal Languages:** Defining Grammar, Derivations, Languages generated by Grammar, Chomsky Classification of Grammar and Languages, Recursive Enumerable Sets, Operations on Languages, Languages and Automata.	**15**
II	**2.1 Regular Sets and Regular Grammar:** Regular Grammar, Regular Expressions, Finite automata and Regular Expressions, Pumping Lemma and its Applications, Closure Properties, Regular Sets and Regular Grammar **2.2 Context Free Languages:** Context-free Languages, Derivation Tree, Ambiguity of Grammar, CFG simplification, Normal Forms, Pumping Lemma for CFG. **2.3 Pushdown Automata:** Definitions, Acceptance by PDA, PDA and CFG	**15**
III	**3.1 Linear Bound Automata:** The Linear Bound Automata Model, Linear Bound Automata and Languages. **3.2 Turing Machines:** Turing Machine Definition, Representations, Acceptability by Turing Machines, Designing and Description of Turing Machines, Turing Machine Construction, Variants of Turing Machine. **3.3 Undecidability:** The Church-Turing thesis, Universal Turing Machine, Halting Problem, Introduction to Unsolvable Problems.	**15**

Tutorials:

1. Problems on generating languages for given simple grammar.
2. Problems on DFA and NDFA equivalence.
3. Problems on generating Regular Expressions.
4. Problems on drawing transition state diagrams for Regular Expressions.
5. Problems on Regular Sets and Regular Grammar.
6. Problems on Ambiguity of Grammar.
7. Problems on working with PDA.
8. Problems on working with Turing Machines.
9. Problems on generating derivation trees.
10. Problems on Linear Bound Automata/Universal Turing Machine.

Self-Learning topics

- Problem Solving Exercises based on the syllabus topics from unit no. I, II and III.

References: Paper-I US-SCS-301 Theory of Computation

Textbook(s)	Theory of Computer Science, K. L. P Mishra, Chandrasekharan, PHI,3rd Edition. Introduction to Computer Theory, Daniel Cohen, Wiley,2nd Edition. Introductory Theory of Computer Science, E.V. Krishnamurthy,Affiliated East-West Press.
Additional Reference(s)	Theory of Computation, Kavi Mahesh, Wiley India. Elements of The Theory of Computation, Lewis, Papadimitriou, PHI. Introduction to Languages and the Theory of Computation, John E Martin, McGraw-Hill Education. Introduction to Theory of Computation, Michel Sipser, Thomson.
Online Reference(s)	**1.** https://nptel.ac.in/courses/111/103/111103016/ https://onlinecourses.nptel.ac.in/noc21_cs83/preview https://www.edx.org/course/automata-theory

Course Code: US-SCS-302 Core Java

Unit No.	Content	No. of Lectures
I	**1.1 The Java Language:** Features of Java, Java programming format, Java Tokens, Java Statements, Java Data Types, Typecasting, Arrays **1.2 OOPS:** Introduction, Class, Object, Static Keywords, Constructors, this Key Word, Inheritance, super Key Word, Polymorphism (overloading and overriding), Abstraction, Encapsulation, Abstract Classes, Interfaces **1.3 String Manipulations:** String, String Buffer, String Tokenizer **1.4 Packages:** Introduction to predefined packages (java.lang, java.util, java.io, java.sql, java.swing), User Defined Packages, Access specifiers	**15**
II	**2.1 Exception Handling:** Introduction, Pre-Defined Exceptions, Try-Catch-Finally, Throws, throw, User Defined Exception examples	**15**
	2.2 Multithreading: Thread Creations, Thread Life Cycle, Life Cycle Methods, Synchronization, Wait() notify() notify all() methods **2.3 I/O Streams:** Introduction, Byte-oriented streams, Character- oriented streams, File, Random access File, Serialization **2.4 Networking:** Introduction, Socket, Server socket, Client –Server Communication	
III	**3.1 Wrapper Classes:** Introduction, Byte, Short, Integer, Long, Float, Double, Character, Boolean classes **3.2 Collection Framework**: Introduction, util Package interfaces, List, Set, Map, List interface & its classes, Set interface & its classes, Map interface & its classes **3.3 Inner Classes**: Introduction, Member inner class, Static inner class, Local inner class, Anonymous inner class **3.4 AWT:** Introduction, Components, Event-Delegation-Model, Listeners, Layouts, Individual components Label, Button, CheckBox, Radio Button, Choice, List, Menu, Text Field, Text Area	**15**

Self-Learning topics

Sub-Unit	Topic
3.1	**Wrapper Classes:** Introduction, Byte, Short, Integer, Long, Float, Double, Character, Boolean classes.

3.4	**AWT:** Introduction, Components, Event-Delegation-Model, Listeners, Layouts, Individual components Label, Button, CheckBox, Radio Button, Choice, List, Menu, Text Field, Text Area.
- Interview Questions/Programming Exercise(s)/Problem Solving Exercise(s) based on the syllabus topics from unit no. I, II and III.	

References: Paper-II US-SCS-302 Core Java

Textbook(s)	Ierbert Schildt, Java The Complete Reference, Ninth Edition, McGraw-Hill Education, 2014.
Additional Reference(s)	E. Balagurusamy, Programming with Java, Tata McGraw-Hill Education India, 2014. Programming in JAVA, 2nd Ed, Sachin Malhotra & Saurabh Choudhary, Oxford Press. The Java Tutorials: http://docs.oracle.com/javase/tutorial/
Online Reference(s)	https://nptel.ac.in/courses/106/105/106105191/

Course Code: US-SCS-303 Operating Systems

Unit No.	**Content**	**No. of Lectures**
I	**1.1 Introduction and Operating-Systems Structures:** Definition of Operating system, Operating System's role, Operating-System Operations, Functions of Operating System, Computing Environments **1.2 Operating-System Structures:** Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, Operating-System Structure **1.3 Processes**: Process Concept, Process Scheduling, Operations on Processes, Inter-process Communication **1.4 Threads:** Overview, Multicore Programming, Multithreading Models.	**15**
II	**2.1 Process Synchronization:** General structure of a typical process, race condition, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors **2.2 CPU Scheduling:** Basic Concepts, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, SRTF, Priority, RR, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling), Thread Scheduling **2.3 Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock	**15**
II	**3.1 Main Memory:** Background, Logical address space, Physical address space, MMU, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table **32 Virtual Memory:** Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing **33 Mass-Storage Structure:** Overview, Disk Structure, Disk Scheduling, Disk Management **34 File-System Interface:** File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing **35 File-System Implementation:** File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management	**15**

Self-Learning topics

Sub-Unit	Topic
3.3	**Mass-Storage Structure:** Overview, Disk Structure, Disk Scheduling, Disk Management
3.4	**File-System Interface:** File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing
3.5	**File-System Implementation:** File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management
Interview Questions/Programming Exercise(s)/Problem Solving Exercise(s) based on the syllabus topics from unit no. I, II and III.	

References: Paper-III US-SCS-303 Operating Systems

Textbook(s)	braham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley, 8th Edition.
Additional Reference(s)	Achyut S. Godbole, Atul Kahate, Operating Systems, Tata McGraw Hill. Naresh Chauhan, Principles of Operating Systems, Oxford Press. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 4e Fourth Edition, Pearson Education, 2016.
Online Reference(s)	**1.** https://epgp.inflibnet.ac.in/ **2.** https://nptel.ac.in/courses/106/102/106102132/ **3.** https://nptel.ac.in/courses/106/108/106108101/ **4.** https://nptel.ac.in/courses/106/105/106105214/ **5.** https://nptel.ac.in/courses/106/106/106106144/

Course Code: US-SCS-304 Software Engineering

Unit No.	Content	No. of Lectures
I	**1.1 Introduction:** The Nature of Software, Software Engineering, The Software Process, Generic Process Model, The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Component-Based Development, The Unified Process Phases, Agile Development- Agility, Agile Process, Extreme Programming **1.2 Requirement Analysis and System Modeling:** Requirements Engineering, Eliciting Requirements, SRS Validation, Components of SRS, Characteristics of SRS , Object-oriented design using the UML - Class diagram, Object diagram, Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram, Component diagram, Deployment diagram.	15
II	**2.1 System Design:** System/Software Design, Architectural Design, Low-Level Design Coupling and Cohesion, Functional-Oriented Versus The Object-Oriented Approach, Design Specifications, Verification for Design, Monitoring and Control for Design **2.2 Software Measurement and Metrics:** Product Metrics – Measures,	15

	Metrics, and Indicators, Function-Based Metrics, Metrics for Object-Oriented Design, Operation-Oriented Metrics, User Interface Design Metrics, Metrics for Source Code, Halstead Metrics Applied to Testing, Metrics for Maintenance, Cyclomatic Complexity, Software Measurement - Size-Oriented, Function-Oriented Metrics, Metrics for Software Quality.	
III	**3.1 Software Project Management: Estimation** in Project Planning Process –Software Scope And Feasibility, Resource Estimation, Empirical Estimation Models – COCOMO II, Estimation for Agile Development, The Make/Buy Decision, **Project Scheduling** - Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-Line Charts. **3.2 Risk Management** - Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan	**15**

Self-Learning topics

Sub-Unit	**Topic**
1.1	Generic Process Model, The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Component-Based Development, The Unified Process Phases
Interview Questions/Programming Exercise(s)/Problem Solving Exercise(s) based on the syllabus topics from unit no. I, II and III.	

References: Paper-IV US-SCS-304 Software Engineering

Textbook(s)	**1.** Software Engineering, A Practitioner's Approach, Roger S, Pressman.(2014)
Additional Reference(s)	Software Engineering, Ian Sommerville, Pearson Education Software Engineering: Principles and Practices", Deepak Jain, OXFORD University Press Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI Software Engineering: Principles and Practices, Hans Van Vliet, John Wiley & Sons A Concise Introduction to Software Engineering, Pankaj Jalote, Springer
Online Reference(s)	**1.** https://nptel.ac.in/courses/106/105/106105182/ **2.** https://nptel.ac.in/courses/106/105/106105218/ **3.** https://nptel.ac.in/courses/106/101/106101163/

Course Code: US-SCS-305 Physical Computing and IoT Programming

Unit No.	Content	No. of Lectures
I	**SoC and Raspberry Pi** **1.1 System on Chip:** What is System on chip? Structure of System on Chip. **1.2 SoC products:** FPGA, GPU, APU, Compute Units. **1.3 ARM 8 Architecture:** SoC on ARM 8. ARM 8 Architecture Introduction **1.4 Introduction to Raspberry Pi:** Introduction to Raspberry Pi, Raspberry Pi Hardware, Preparing your raspberry Pi. **1.5 Raspberry Pi Boot:** Learn how this small SoC boots without BIOS. Configuring boot sequences and hardware.	15
II	**Programming Raspberry Pi** **2.1 Raspberry Pi and Linux:** About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands **2.2 Programing interfaces:** Introduction to Node.js, Python. **2.3 Raspberry Pi Interfaces:** UART, GPIO, I2C, SPI **2.4 Useful Implementations:** Cross Compilation, Pulse Width Modulation, SPI for Camera.	15
III	**3.1 Introduction to IoT:** What is IoT? IoT examples, Simple IoT LED Program. **IoT and Protocols** **3.2 IoT Security:** HTTP, UPnp, CoAP, MQTT, XMPP**.** **3.3 IoT Service as a Platform:** Clayster, Thinger.io, SenseIoT, carriots and Node RED. **3.4 IoT Security and Interoperability:** Risks, Modes of Attacks, Tools for Security and Interoperability.	15

Self-Learning topics

Sub-Unit	Topic
2.1	**Raspberry Pi and Linux:** About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands
2.2	**Programing interfaces:** Introduction to Node.js, Python. - Interview Questions/Programming Exercise(s)/Problem Solving Exercise(s) based on the syllabus topics from unit no. I, II and III.

References: Paper - V US-SCS-305 Physical Computing and IoT Programming

Textbook(s)	Learning Internet of Things, Peter Waher, Packt Publishing(2015) Mastering the Raspberry Pi, Warren Gay, Apress(2014)
Additional Reference(s)	**1.** Abusing the Internet of Things, Nitesh Dhanjani, O'Reilly
Online Reference(s)	**1.** https://nptel.ac.in/courses/106/105/106105166/

Course Code: US-SCS-306 PHP with E-Commerce

Jnit No.	Content	No. of Lectures
I	1.1 **PHP-** Variables and Operators, Program Flow, Arrays, Working with Files and Directories, Working with Databases, MySQLi extension (the "i" stands for improved), Working with Cookies, Sessions and Headers. 1.2 **PHP OOP-** Classes & Objects, Constructor, Destructor, Access Modifiers, Inheritance, Class Constants, Abstract Classes, Interfaces, Traits, Static Methods, Static Properties, PHP Namespaces, PHP Iterables.	15
II	2.1 **PHP & MySQL Database using MySQLi and PHP Data Objects (PDO)-** Connect to MySQL, Create a MySQL Database, Create Table, Insert Data, Get Last Inserted ID, Insert Multiple Records, Prepared Statements, Select Data, Where Clause, Order By Clause, Delete Data, Update Data, Limit Data. 2.2 **Laravel:** Getting Started, Architecture Concepts, The Basics- Routing, Middleware, CSRF Protection, Controllers, Requests, Responses, Views, Blade Templates, URL Generation, Session, Validation, Error Handling & Logging, Digging Deeper- Artisan Console, Contracts, Events, File Storage, Localization, Mail, Notifications, Security- Authentication, Authorization, Email Verification, Encryption, Hashing, Password Reset, Working with Databases.	15
III	3.1 **Introduction to Ecommerce:** What is Electronic Commerce? Benefits of Electronic Commerce. Classification Electronic Commerce.	15
	Applications of Electronic Commerce Technologies. Taxonomy of Internet Commerce Business Models. 3.2 What is EDI? Building Blocks of EDI Systems (term definitions), Electronic Payment Systems. Business Models for Ecommerce Revenue Model, Marketing on the web: Internet based Advertisement, Website usability, consumer-oriented e-commerce. 3.3 Installing WooCommerce/Magento, General Settings in WooCommerce/Magento - Payment Gateways, Shipping, Taxes, Emails, Manging Products, Plugins/Extensions.	

Self-Learning topics

Sub-Unit	Topic
3.2	What is EDI? Building Blocks of EDI Systems (term definitions), Electronic Payment Systems. Business Models for Ecommerce Revenue Model, Marketing on the web: Internet based Advertisement, Website usability, consumer-oriented e-commerce.
3.3	Study of Open Source E-Commerce Platforms.
Interview Questions/Programming Exercise(s)/Problem Solving Exercise(s) based on the syllabus topics from unit no. I, II and III.	

References: Paper - VI US-SCS-306 PHP with E-Commerce

Textbook(s)	HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press Web Programming and Interactive Technologies, scriptDemics, StarEdu Solutions India. PHP: A Beginners Guide, Vikram Vaswani, TMH Electronic Commerce: Framework Technologies & Applications, Bharat Bhasker, TMH Ecommerce, Gary P. Schneider, Cengage Learning
Additional Reference(s)	Learning PHP, MySQL, JavaScript, CSS & HTML5, Robin Nixon, O'Reilly. PHP, MySQL, JavaScript & HTML5 All-in-one for Dummies, Steve Suehring, Janet Valade Wiley.
Online Reference(s)	https://devdocs.magento.com/ https://docs.woocommerce.com/ https://www.w3schools.com/

Course Code: US-SCS-307 Android Developer Fundamentals

Unit No.	Content	No. of Lectures
I	**Introduction to Android** **1.1** What is Android?, Build first Android app, Android Studio, Layouts and resources for the UI, Text and scrolling views **1.2 Activities and intents:** Activity lifecycle and state, Implicit intents **1.3 Testing, debugging, and using support libraries:** The Android Studio debugger, C App testing, The Android Support Library	**15**
II	**User Experience** **2.1 User interaction:** Buttons and clickable images, Input controls, Menus and pickers, User navigation, Recycler View **2.2 Delightful user experience:** Drawables, styles, and themes, Material Design, Resources for adaptive layouts **2.3 Testing your UI:** UI testing	**15**
III	**Working in the background** **3.1 Background tasks:** AsyncTask and AsyncTaskLoader, Internet connection, Broadcast receivers, Services **3.2 Alarms and schedulers:** Notifications, Alarms, Efficient data transfer **Saving user data** **3.3 Preferences and settings:** Data storage, Shared preferences, App settings **3.4 Storing data with Room:** SQLite primer, Room, LiveData, and ViewModel	**15**

Self-Learning topics

Sub-Unit	Topic
2.2	**Delightful user experience:** Drawables, styles, and themes, Material Design, Resources for adaptive layouts
2.3	**Testing your UI:** UI testing
Interview Questions/Programming Exercise(s)/Problem Solving Exercise(s) based on the syllabus topics from unit no. I, II and III.	

References: Paper - VII US-SCS-307 Android Developer Fundamentals

Textbook(s)	Beginning Android 4 Application Development", Wei-Meng Lee, March 2012, WROX.
Additional Reference(s)	https://developer.android.com/docs https://developer.android.com/codelabs/build-your-first-android-app#0 https://developer.android.com/courses/fundamentals-training/overview-v2
Online Reference(s)	**1.** https://nptel.ac.in/courses/106/106/106106147/

Part - 4 Detailed Scheme Practical

Course Code: US-SCS-3P1

Practical I	**US-SCS-302 + US-SCS-303 + US-SCS-304**	**Total Credits: 3**
Unit	**Content**	**No. of Lectures**
1, 2, 3	**US-SCS-302: Core Java** Accept integer values for a, b and c which are coefficients of quadratic equation. Find the solution of quadratic equation. Accept two n x m matrices. Write a Java program to find addition of these matrices. Accept n strings. Sort names in ascending order. Create a package: Animals. In package animals create interface Animal with suitable behaviors. Implement the interface Animal in the same package animals. Demonstrate Java inheritance using extends keyword. Demonstrate method overloading and method overriding in Java. Demonstrate creating your own exception in Java. Using various swing components design Java application to accept a student's resume. (Design form) Write a Java List example and demonstrate methods of Java List interface. Design simple calculator GUI application using AWT components.	**45**
1, 2, 3	**US-SCS-303: Operating Systems** *Practical can be implemented either in JAVA or any other programming language.* **1. Process Communication:** Give solution to the producer–consumer problem using shared memory. Give solution to the producer–consumer problem using message passing. One form of communication in a Client–Server Systems environment is Remote method invocation (RMI). RMI is a Java feature similar to RPCs. RMI allows a thread to invoke a method on a remote object. Objects are considered remote if	**45**

	they reside in a different Java virtual machine (JVM). Demonstrate RMI program for adding/subtracting/multiplying/dividing two numbers. **2. Threads:** The Java version of a multithreaded program that determines the summation of a non-negative integer. The Summation class implements the Runnable interface. Thread creation is performed by creating an object instance of the Thread class and passing the constructor a Runnable object. Write a multithreaded Java program that outputs prime numbers. This program should work as follows: The user will run the program and will enter a number on the command line. The program will then create a separate thread that outputs all the prime numbers less than or equal to the number entered by the user. The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5. 8, ... Formally, it can be expressed as: $fib_0 = 0$, $fib_1 = 1$, $fib_n = fib_{n-1} + fib_{n-2}$ Write a multithreaded program that generates the Fibonacci sequence using either the Java. **3. Synchronization:** Give Java solution to Bounded buffer problem. Give solution to the readers–writers problem using Java synchronization. The Sleeping-Barber Problem: A barber shop consists of awaiting room with *n* chairs and a barber room with one barber chair. If there are no customers to be served, the barber goes to sleep. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy but chairs are available, then the customer sits in one of the free chairs. If the barber is asleep, the customer wakes up the barber. Write a program to coordinate the barber and the customers using Java synchronization. Implement FCFS scheduling algorithm in Java. Implement SJF (with no preemption) scheduling algorithm in Java. Implement RR scheduling algorithm in Java. Write a Java program that implements the banker's algorithm Write a Java program that implements the FIFO page-replacement algorithm. Write a Java program that implements the LRU page-replacement algorithm. Design a File System in Java.	
1, 2, 3	**US-SCS-304: Software Engineering** Do the following exercises for any one project/case study. Perform the project development in team of 3 - 5 members. Draw the UML diagrams using StarUML/ SmartDraw / Visual Paradigm / any relevant online tool. Preparing Software Requirements Specifications (Event List and Event Table). Modeling Entity Relationship Diagrams. Modeling UML Class Diagrams. Modeling UML Use Case Diagrams and Capturing Use Case Scenarios. Modeling Sequence/Collaboration diagrams. Modeling State Transition Diagrams. Modeling Activity diagram. Modeling Component diagram. Modeling Deployment diagram. Preparing Issue Tracker Sheet using MS Excel.	**45**

	Sample Projects/Case Studies: Passport automation System Restaurant Business Model Online Exam Registration Online Shopping Online course reservation system E-ticketing Software Personnel Management System E-book management System Recruitment system	

Course Code: US-SCS-3P2

ctical II	**US-SCS-305 + US-SCS-306 + US-SCS-307**	**Total Credits: 3**
Unit	**Content**	**No. of Lectures**
1, 2, 3	**US-SCS-305: Physical Computing and IoT Programming** Preparing Raspberry Pi: Hardware preparation and Installation. Linux Commands: Exploring the Raspbian. GPIO: Light the LED with Python GPIO: LED Grid Module: Program the 8X8 Grid with Different Formulas SPI: Camera Connection and capturing Images using SPI Real Time Clock display using PWM. Stepper Motor Control: PWM to manage stepper motor speed. Node RED: Connect LED to Internet of Things Stack of Raspberry Pi for better Computing and analysis Create a simple Web server using Raspberry Pi	**45**
1, 2, 3	**US-SCS-306: PHP with E-Commerce** Write PHP scripts for Retrieving data from HTML forms Performing certain mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number Working with Arrays Working with Files (Reading / Writing) Write PHP scripts for (using MySQLi and PHP Data Objects (PDO))- Working with Databases (Storing Records / Reprieving Records and Display them) Storing and Retrieving Cookies Storing and Retrieving Sessions Web Application based on Laravel with database operations.	**45**

	Write programs that interacts with an HTML forms and performs submission at the server side using OOP with PHP. (classes/objects, constructor, constants, static methods and properties) Program to create product, order and demonstrate an Ecommerce business flow using Magento/WooCommerce.	
1, 2, 3	**US-SCS-307: Android Developer Fundamentals** Install Android Studio and Run Hello World Program. Create an android app with Interactive User Interface using Layouts. Create an android app that demonstrates working with TextView Elements. Create an android app that demonstrates Activity Lifecycle and Instance State. Create an android app that demonstrates the use of Keyboards, Input Controls, Alerts, and Pickers. Create an android app that demonstrates the use of an Options Menu. Create an android app that demonstrate Screen Navigation Using the App Bar and Tabs. Create an android app to Connect to the Internet and use BroadcastReceiver. Create an android app to show Notifications and Alarm manager. Create an android app to save user data in a database and use of different queries.	**45**

Part - 5 The Scheme of Teaching and Examination is as under: Second Year Semester – IV

Summary

Sr. No.	Choice Based Credit System			Subject Code	Remarks
1	Core Course (Computer Science)			US-SCS-401 US-SCS-402 US-SCS-403 US-SCS-404	
2	Elective Course	Discipline Specific Elective (DSE) Course			
		2.1	Interdisciplinary Specific Elective (IDSE) Course		
		2.2	Dissertation/Project		
		2.3	Generic Elective (GE) Course	US-SCS-405	
3	Ability Enhancement Courses (AEC)				
4	Skill Enhancement Courses (SEC)			US-SCS-406 US-SCS-407	

Second Year Semester -IV Internal and External Detailed Evaluation Scheme

Sr. No.	Subject Code	Subject Title	Periods Per Week					Credits	Internals (40)				Total Marks
			Units	S.L.	L	T	P		SLE	CT + AT = 15 + 5	PA	SEE	
1	US-SCS-401	Design and Analysis of Algorithms	3	20% *	3	0	-	2	10	20	10	60	100
2	US-SCS-402	Advanced Java	3	20% *	3	0	-	2	10	20	10	60	100
3	US-SCS-403	Computer Networks	3	20% *	3	0	-	2	10	20	10	60	100
4	US-SCS-404	Software Testing and Quality Assurance	3	20% *	3	0	-	2	10	20	10	60	100
5	US-SCS-405	Linear Algebra using Python	3	20% *	3	0	-	2	10	20	10	60	100
6	US-SCS-406	. NET Technologies	3	20% *	3	0	-	2	10	20	10	60	100
7	US-SCS-407	Introduction to Research Methodology	3	20% *	3	0	-	2	10	20	10	60	100
8	US-SCS-4P1	Practical of US-SCS-401 + US-SCS-402 + US-SCS-403	-	-	-	-	6	3				150 (90+ 60)	150
9	US-SCS-4P2	Practical of US-SCS-404 + US-SCS-405 + US-SCS-406	-	-	-	-	6	3				150 (90+ 60)	150
	Total Lectures/ Credits							**20**	**Total Marks**				**1000**

****One to two lectures to be taken for CONTINUOUS self –learning Evaluation.**

Second Year Semester – IV Units – Topics – Teaching Hours

Sr. No.	Subject Code & Title	Subject Unit Title		Hours/ Lectures	Total No. of hours/ lectures	Credit	Total Marks
1	US-SCS-401 Design and Analysis of Algorithms	1	Foundations, Sorting & Order Statistics	15	45L	2	100 (60+40)
		2	Trees, Introduction to Graphs, Graph Algorithms	15			
		3	Algorithm Design Techniques- Divide & Conquer, Greedy Algorithms, Dynamic Programming	15			
2	US-SCS-402 Advanced Java	1	Swing, JDBC	15	45L	2	100 (60+40)
		2	Servlets, JSP	15			
		3	Java Beans, Struts2, JSON	15			
3	US-SCS-403 Computer Networks	1	Introduction to Network Models	15	45L	2	100 (60+40)
		2	Introduction to Physical Layer and Data Link Layer	15			
		3	Network Layer, Transport Layer	15			
4	US-SCS-404 Software Testing and Quality Assurance	1	Software Testing and Introduction to quality, Verification and Validation, Software Testing Techniques	15	45L	2	100 (60+40)
		2	Software Testing Strategies, Software Metrics, Defect Management	15			
		3	Software Quality Assurance, Quality Improvement, Quality Costs	15			
5	US-SCS-405 Linear Algebra using Python	1	Matrices, Python Libraries for Linear Algebra	15	45L	2	100 (60+40)
		2	Vector Spaces, Orthogonality	15			
		3	Determinants, Eigenvalues and Eigenvectors, Principal Component Analysis, Singular Value Decomposition	15			
6	US-SCS-406 .NET Technologies	1	Introduction to .NET, C# Language Basics, ASP.NET, Web Controls	15	45L	2	100 (60+40)
		2	Validation, State Management, Data Access - ADO.NET, LINQ, Entity Framework	15			
		3	Introduction to ASP.NET Core, ASP.NET Core Web App, Web APIs with ASP.NET Core	15			

7	US-SCS-407 Introduction to Research Methodology	1	Introduction to Research Methodology, Defining the Research Problem, Formulating a Research Problem	15	45L	2	100 (60+40)
		2	The Research Design, Tools for Data Collection, Sampling, Hypothesis Testing	15			
		3	Technical Writing, Preparing the Text, Preparing the Tables and Figures, Publishing the Paper	15			
8	US-SCS-4P1	1	Practical based on US-SCS-401	3	45x3= 135 lectures per batch	2	150 (90+60)
		2	Practical based on US-SCS-402	3			
		3	Practical based on US-SCS-403	3			
9	US-SCS-4P2	1	Practical based on US-SCS-404	3	45x3= 135 lectures per batch	2	100 (90+60)
		2	Practical based on US-SCS-405	3			
		2	Practical based on US-SCS-406	3			
			TOTAL			**20**	**1000**

- **Lecture Duration – 48 Minutes**
- **One Credit =15 hours**

L: Lecture: Tutorials P: Practical Ct-Core Theory, Cp-Core Practical, SLE- Self learning evaluation CT-Commutative Test, SEE- Semester End Examination, PA- Project Assessment, AT- Attendance

Part - 6 Detailed Scheme Theory

Curriculum Topics along with Self-Learning topics - to be covered, through self-learning mode along with the respective Unit. Evaluation of self-learning topics to be undertaken before the concluding lecture instructions of the respective UNIT

Course Code: US-SCS-401 Design and Analysis of Algorithms

Unit No.	Content	No. of Lectures
I	**1.1 Foundations:** Role of Algorithms in Computing, Analyzing Algorithms, How to Compare Algorithms?, Rate of Growth, Commonly used Rate of Growth, Types of Analysis, Asymptotic Notations- Big Oh, Big Omega, Big Theta, Properties of notations, Commonly used logarithms and summations, Performance characteristics of algorithms, divide-and-conquer approach, analyzing divide-and-conquer algorithms, Recurrences- The substitution method, The recursion-tree method, The master method. **1.2 Sorting and Order Statistics:** Heapsort- Heaps, Maintaining the heap property, building a heap, the heapsort algorithm, Priority queues, Quicksort- Description of quicksort, Performance of quicksort, A randomized version of quicksort, Analysis of quicksort, Sorting in	**15**

	Linear Time- Lower bounds for sorting, counting sort, Radix sort, Bucket sort, Medians and Order Statistics- Minimum and maximum, Selection in expected linear time, Selection in worst-case linear time.	
II	2.1 **Trees:** What is a Tree? Glossary, Binary Trees, Types of Binary Trees, Properties of Binary Trees, Binary Tree Traversals, Generic Trees (N-ary Trees), Threaded Binary Tree Traversals, Expression Trees, Binary Search Trees (BSTs), Balanced Binary Search Trees, AVL (Adelson-Velskii and Landis) Trees, Red Black Trees, B-Trees. 2.2 **Introduction to Graphs:** Basic notation and terminology for graphs, Some special simple graphs, bipartite graphs & matchings, complete bipartite graphs, Representing Graphs, Graph Isomorphism, Graph Coloring. 2.3 **Graph Algorithms:** Minimum weight spanning trees, Kruskal's algorithm, Prim's algorithm, Comments on efficiency, Dijkstra's algorithm for shortest paths, Bellman Ford Algorithm, The Travelling Salesman Problem, Cycle detection, Strongly connected components.	15
III	**Algorithm Design Techniques** **3.1 Divide and Conquer Algorithms:** Introduction, What is Divide and Conquer Strategy? Divide and Conquer Visualization, Understanding Divide and Conquer, Advantages of Divide and Conquer, Disadvantages of Divide and Conquer, Master Theorem, Divide and Conquer Applications. 3.2 **Greedy Algorithms:** Introduction, Greedy Strategy, Knapsack Problem, Huffman Codes, Job Sequencing with Deadlines. 3.3 **Dynamic Programming:** Introduction, What is Dynamic Programming Strategy? Properties of Dynamic Programming Strategy, Problems which can be solved using Dynamic Programming, Dynamic Programming Approaches, Examples of Dynamic Programming Algorithms, Understanding Dynamic Programming, Longest Common Subsequence, Matrix Chain Multiplication, 0/1 Knapsack, The Subset Sum Problem, Multistage graphs, All Pairs Shortest Paths, The Travelling Salesman Problem.	15

Self-Learning topics

Sub-Unit	Topic
2.3	**Graph Algorithms:** Minimum weight spanning trees, Kruskal's algorithm, Prim's algorithm, Comments on efficiency, Dijkstra's algorithm for shortest paths, Bellman Ford Algorithm, The Travelling Salesman Problem, Cycle detection, Strongly connected components.
- Interview Questions/Programming Exercise(s)/Problem Solving Exercise(s) based on the syllabus topics from unit no. I, II and III.	

References: Paper-I US-SCS-401 Design and Analysis of Algorithms

Textbook(s)	Data Structure and Algorithmic Thinking with Python, Narasimha Karumanchi, CareerMonk Publications, 2016. Introduction to Algorithm, Thomas H Cormen, PHI.
Additional Reference(s)	Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, 2016, Wiley. Fundamentals of Computer Algorithms, Sartaj Sahni and Sanguthevar Rajasekaran

	Ellis Horowitz, Universities Press
Online Reference(s)	**1.** https://nptel.ac.in/courses/106/106/106106145/ **2.** https://nptel.ac.in/courses/106/105/106105157/ **3.** https://nptel.ac.in/courses/106/105/106105164/

Course Code: US-SCS-402 Advanced Java

Unit No.	Content	No. of Lectures
I	**1.1 Swing:** Need for swing components, Difference between AWT and swing, Components hierarchy, Panes, Swing components: Jlabel, JTextField and JPasswordField, JTextAres, JButton, JCheckBox, JRadioButton, JComboBox and JList. **1.2 JDBC:** Introduction, JDBC Architecture, Types of Drivers, Statement, ResultSet, Read Only ResultSet, Updatable ResultSet, Forward Only ResultSet, Scrollable ResultSet, PreparedStatement, Connection Modes, SavePoint, Batch Updations, CallableStatement, BLOB & CLOB.	15
II	**2.1 Servlets:** Introduction, Web application Architecture, Http Protocol & Http Methods, Web Server & Web Container, Servlet Interface, GenericServlet, HttpServlet, Servlet Life Cycle, ServletConfig, ServletContext, Servlet Communication, Session Tracking Mechanisms **2.2 JSP:** Introduction, JSP LifeCycle, JSP Implicit Objects & Scopes, JSP Directives, JSP Scripting Elements, JSP Actions: Standard actions and customized actions,	15
III	**3.1 Java Beans:** Introduction, JavaBeans Properties, Examples **3.2 Struts 2:** Basic MVC Architecture, Struts 2 framework features, Struts 2 MVC pattern, Request life cycle, Examples, Configuration Files, Actions, Interceptors, Results & Result Types, Value Stack/OGNL **3.3 JSON:** Overview, Syntax, DataTypes, Objects, Schema, Comparison with XML, JSON with Java	15

Self-Learning topics

1.1	**Swing:** Need for swing components, Difference between AWT and swing, Components hierarchy, Panes, Swing components: Jlabel, JTextField and JPasswordField, JTextAres, JButton, JCheckBox, JRadioButton, JComboBox and JList.
Interview Questions/Programming Exercise(s)/Problem Solving Exercise(s) based on the syllabus topics from unit no. I, II and III.	

References: Paper-II US-SCS-402 Advanced Java

Textbook(s)	Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II–Advanced Features Prentice Hall PTR,9th Edition Herbert Schildt, Java2: The Complete Reference, Tata McGraw-Hill,5th Edition Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD) ,3rd Edition
Additional Reference(s)	Advanced Java Programming, Uttam K. Roy, Oxford University Press *The Java Tutorials: http://docs.oracle.com/javase/tutorial/)* The Java Tutorials of Sun Microsystems Inc
Online Reference(s)	**1.** https://nptel.ac.in/courses/106/105/106105191/ **2.** https://javaee.github.io/tutorial/toc.html

Course Code: US-SCS-403 Computer Networks

Unit No.	Content	No. of Lectures
I	**Introduction Network Models:** **1.1** Introduction to data communication, Components, Data Representation, Data Flow, Networks, Network Criteria, Physical Structures, Network types, Local Area Network, Wide Area Network, Switching, The Internet, Accessing the Internet, standards and administration Internet Standards. **1.2** Network Models, Protocol layering, Scenarios, Principles of Protocol Layering, Logical Connections, TCP/IP Protocol Suite, Layered Architecture, Layers in the TCP/IP Protocol Suite, Encapsulation and Decapsulation, Addressing, Multiplexing and Demultiplexing. Detailed introduction to Physical Layer, Detailed introduction to Data-Link Layer, Detailed introduction to Network Layer, Detailed introduction to Transport Layer, Detailed introduction to Application Layer. **1.3** Data and Signals, Analog and Digital Data, Analog and Digital Signals, Sine Wave Phase, Wavelength, Time and Frequency Domains, Composite Signals, Bandwidth, Digital Signal, Bit Rate, Bit Length, Transmission of Digital Signals, Transmission Impairments, Attenuation, Distortion, Noise, Data Rate Limits, Performance, Bandwidth, Throughput, Latency (Delay)	**15**
II	**Introduction to Physical Layer and Data-Link Layer:** **2.1** Digital Transmission digital-to-digital conversion, Line Coding, Line Coding Schemes, analog-to-digital conversion, Pulse Code Modulation (PCM), Transmission Modes, Parallel Transmission, Serial Transmission.	**15**
	Analog Transmission, digital-to-analog Conversion, Aspects of Digital-to- Analog Conversion, Amplitude Shift Keying, Frequency Shift Keying, Phase Shift Keying, analog-to-analog Conversion, Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM), Multiplexing, Frequency-Division Multiplexing, Wavelength-Division Multiplexing, Time-Division Multiplexing. Transmission Media, Guided Media, Twisted- Pair Cable, Coaxial Cable, Fiber-Optic Cable. Switching, Three Methods of Switching , Circuit Switched Networks, Packet Switching, **2.2** Introduction to Data-Link Layer, Nodes and Links, Services, Two Sub-layers, Three Types of addresses, Address Resolution Protocol (ARP). Error Detection and Correction, introduction, Types of Errors, Redundancy, Detection versus Correction,	
III	**Network layer, Transport Layer** **3.1** Media Access Control (MAC), random access, CSMA, CSMA/CD, CSMA/CA, controlled access, Reservation, Polling, Token Passing, channelization, FDMA, TDMA, CDMA. **3.2** Connecting Devices and Virtual LANs, connecting devices, Hubs, Link-Layer Switches, Routers. **3.3** Introduction to Network Layer, network layer services, Packetizing, Routing and Forwarding, Other Services, IPv4 addresses, Address Space, Classful Addressing.	**15**

	3.4 Unicast Routing, General Idea, Least-Cost Routing, Routing Algorithms, Distance-Vector Routing, Link-State Routing, Path-Vector Routing, Introduction to Transport Layer, Transport-Layer Services, Connectionless and Connection-Oriented Protocols. **3.5** Transport-Layer Protocols, Service, Port Numbers, User Datagram Protocol, User Datagram, UDP Services, UDP Applications, Transmission Control Protocol, TCP Services, TCP Features, Segment.	

Self-Learning topics

3.2	Connecting Devices and Virtual LANs, connecting devices, Hubs, Link-Layer Switches, Routers.
3.5	Transport-Layer Protocols, Service, Port Numbers, User Datagram Protocol, User Datagram, UDP Services, UDP Applications, Transmission Control Protocol, TCP Services, TCP Features, Segment.
Interview Questions/Programming Exercise(s)/Problem Solving Exercise(s) based on the syllabus topics from unit no. I, II and III.	

References: Paper-III US-SCS-403 Computer Networks

Textbook(s)	Data Communications and Networking, Behrouz A. Forouzan, Fifth Edition, TMH, 2013. Computer Network, Andrew S. Tanenbaum, David J. Wetherall, Fifth Edition, Pearson Education, 2011.
Additional Reference(s)	Computer Network, Bhushan Trivedi, Oxford University Press Data and Computer Communication, William Stallings, PHI
Online Reference(s)	**1.** https://nptel.ac.in/courses/106/105/106105080/ **2.** https://nptel.ac.in/courses/106/105/106105081/ **3.** https://nptel.ac.in/courses/106/106/106106091/ **4.** https://nptel.ac.in/courses/106/105/106105183/

Course Code: US-SCS-404 Software Testing and Quality Assurance

Unit No.	Content	No. of Lectures
I	**1.1 Software Testing and Introduction to quality:** Introduction, Nature of errors, an example for Testing, Definition of Quality, QA, QC, QM and SQA, Software Development Life Cycle, Software Quality Factors. **1.2 Verification and Validation:** Definition of V & V, Different types of V & V Mechanisms, Concepts of Software Reviews, Inspection and Walkthrough. **1.3 Software Testing Techniques:** Testing Fundamentals, Test Case Design, White Box Testing and its types, Black Box Testing and its types.	**15**
II	**2.1 Software Testing Strategies:** Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing **2.2 Software Metrics:** Concept and Developing Metrics, Different types of Metrics, Complexity metrics **2.3 Defect Management:** Definition of Defects, Defect Management Process, Defect Reporting, Metrics Related to Defects, Using Defects for Process Improvement.	**15**
III	**3.1 Software Quality Assurance:** Quality Concepts, Quality Movement, Background Issues, SQA activities, Software Reviews, Formal Technical	**15**

	Reviews, Formal approaches to SQA, Statistical Quality Assurance, Software	
	Reliability, The ISO 9000 Quality Standards, , SQA Plan , Six sigma, Informal Reviews. **3.2 Quality Improvement:** Introduction, Pareto Diagrams, Cause-effect Diagrams, Scatter Diagrams, Run charts **3.3 Quality Costs:** Defining Quality Costs, Types of Quality Costs, Quality Cost Measurement, Utilizing Quality Costs for Decision-Making.	

Self-Learning topics

2.2	**Software Metrics:** Concept and Developing Metrics, Different types of Metrics, Complexity metrics
3.2	**Quality Improvement:** Introduction, Pareto Diagrams, Cause-effect Diagrams, Scatter Diagrams, Run charts
Interview Questions/Programming Exercise(s)/Problem Solving Exercise(s) based on the syllabus topics from unit no. I, II and III.	

References: Paper-IV US-SCS-404 Software Testing and Quality Assurance

Textbook(s)	Software Engineering for Students, A Programming Approach, Douglas Bell, 4 th Edition, Pearson Education, 2005 Software Engineering – A Practitioners Approach, Roger S. Pressman, 5 th Edition, Tata McGraw Hill, 2001 Quality Management, Donna C. S. Summers, 5 th Edition, Prentice-Hall, 2010. Total Quality Management, Dale H. Besterfield, 3 rd Edition, Prentice Hall, 2003.
Additional Reference(s)	Software engineering: An Engineering approach, J.F. Peters, W. Pedrycz , John Wiley,2004 Software Testing and Quality Assurance Theory and Practice, Kshirsagar Naik, Priyadarshi Tripathy , John Wiley & Sons, Inc. , Publication, 2008 Software Engineering and Testing, B. B. Agarwal, S. P. Tayal, M. Gupta, Jones and Bartlett Publishers, 2010.
Online Reference(s)	**1.** https://nptel.ac.in/courses/106/101/106101163/

Course Code: US-SCS-405 Linear Algebra using Python

Unit No.	Content	No. of Lectures
I	**1.1 Matrices:** Introduction to Vectors and Matrices, The Geometry of Linear Equations, Matrix Notation and Matrix Multiplication, Transposes, Inverses, Gaussian Elimination, factorization $A=LU$, rank, Inverses and Transposes, Special Matrices and Applications. **1.2 Python Libraries for Linear Algebra:** numpy, scipy, matplotlib, seaborn	**15**
II	**2.1 Vector Spaces:** Vector Spaces and Subspaces, Solving $Ax = 0$ and $Ax = b$, Linear Independence, Basis, and Dimension, The Four Fundamental Subspaces, Linear Transformations. **2.2 Orthogonality:** Orthogonal Vectors and Subspaces, Cosines and Projections onto Lines, Projections and Least Squares, Orthogonal Bases and Gram-Schmidt.	**15**
III	**3.1 Determinants:** Introduction, Properties of the Determinant, Formulas for the Determinant, Applications of Determinants.	**15**

	3.2 Eigenvalues and Eigenvectors: Characteristic polynomial, Computing eigenvalues and eigenvectors, Change of Basis, Diagonalization of a matrix, Computing powers of A. **3.3** Principal Component Analysis, Singular Value Decomposition.	

Self-Learning topics

1.2	**Python Libraries for Linear Algebra:** numpy, scipy, matplotlib, seaborn
Interview Questions/Programming Exercise(s)/Problem Solving Exercise(s) based on the syllabus topics from unit no. I, II and III.	

References: Paper-V US-SCS-405 Linear Algebra using Python

Textbook(s)	inear Algebra and Its Applications, Gilbert Strang, Cengage Learning, 4th Edition (2007).
Additional Reference(s)	**1.** Coding the Matrix Linear Algebra through Applications to Computer Science Edition 1, PHILIP N. KLEIN, Newtonian Press (2013) **2.** Linear Algebra and Probability for Computer Science Applications, Ernest Davis, A K Peters/CRC Press (2012). **3.** Linear Algebra and Its Applications, David C Lay, Pearson Education India; 3rd Edition (2002) **4.** https://numpy.org/ **5.** https://www.scipy.org/ **6.** https://matplotlib.org/ **7.** https://seaborn.pydata.org/
Online Reference(s)	**1.** https://nptel.ac.in/courses/111/104/111104137/ **2.** https://nptel.ac.in/courses/108/106/108106171/ **3.** https://nptel.ac.in/courses/111/106/111106135/

Course Code: US-SCS-406 .NET Technologies

Unit No.	Content	No. of Lectures
I	**1.1 Introduction to .NET** - Cross platform, Open source, .NET terminology - .NET Core, .NET Framework, Tools and productivity - Programming languages, IDEs, SDK and runtimes, .NET architectural components, Execution Model - CLR, Managed Execution Process, Assemblies in .NET, .NET application publishing overview. **1.2 C# Language Basics:** Console Application, Variables and Data Types, Comments, Conditional Logic, Loops, Methods, Classes, Value Types and Reference Types, Namespaces and Assemblies, Inheritance, Static Members, Casting Objects, Partial Classes. **1.3 ASP.NET:** Creating Websites, Anatomy of a Web Form - Page Directive, Doctype, Writing Code - Code-Behind Class, Adding Event Handlers, Anatomy of an ASP.NET Application - ASP.NET File Types, ASP.NET Web Folders. **1.4 Web Controls** - View State, Page Class, global.asax File, web.config File, Web Control Classes, WebControl Base Class, List Controls, Table Controls, Web Control Events and AutoPostBack, Page Life Cycle	**15**
II	**2.1 Validation:** Validation Controls, Server-Side Validation, Client-Side Validation, HTML5 Validation, Manual Validation, Validation with Regular Expressions **2.2 State Management:** ViewState, Cross-Page Posting, Query String,	**15**

	Cookies, Session State, Configuring Session State, Application State **2.3 Data Access - ADO.NET:** Creating a Connection, Select Command, DataReader, Disconnected Data Access, **Data Controls:** GridView, DetailsView, FormView.	
	2.4 LINQ: LINQ: What is LINQ, Types of LINQ, Query expression basics - from, select, group, where, orderby, join, let, into, Query variable, Query a collection of objects, Aggregate functions, Partitioning Operators - Skip, Take, Quantifier Operators - All, Any, Contains, Element Operators (Methods) - ElementAt, First, Last, Single, FirstOrDefault **2.5 Entity Framework:** What is Entity Framework?, Entity Framework Features, Entity Data Model, Querying, Saving, Context Class in Entity Framework	
III	**3.1 Introduction to ASP.NET Core**: ASP.NET 4.x and ASP.NET Core, why choose ASP.NET Core, Client-side development, ASP.NET Core target frameworks, .NET Core CLI, NuGet Packages. **3.2 ASP.NET Core Web App** – Scaffolding, MVC App - Model, View, Controller, Razor Pages – Model, Pages folder, wwwroot folder, appsettings.json, Program.cs, Startup.cs, Data Annotations, Entity Framework Core, Packages and Dependencies. **3.3 Web APIs with ASP.NET Core** - RESTful API, ASP.NET Core Web API Project –Web API Controllers, Routing, CRUD operations using HTTP action verbs.	**15**

Self-Learning topics

2.1	**Validation:** Validation Controls, Server-Side Validation, Client-Side Validation, HTML5 Validation, Manual Validation, Validation with Regular Expressions
2.2	**State Management:** ViewState, Cross-Page Posting, Query String, Cookies, Session State, Configuring Session State, Application State
2.3	**Data Access - ADO.NET:** Creating a Connection, Select Command, DataReader, Disconnected Data Access, **Data Controls:** GridView, DetailsView, FormView.
- Interview Questions/Programming Exercise(s)/Problem Solving Exercise(s) based on the syllabus topics from unit no. I, II and III.	

References: Paper-VI US-SCS-406 .NET Technologies

Textbook(s)	**1.** Beginning ASP.NET 4.5 in C#, Matthew MacDonald, Apress(2012)
Additional Reference(s)	The Complete Reference ASP .NET, MacDonald, Tata McGraw Hill Beginning ASP.NET 4 in C# and VB Imar Spanajaars, WROX Professional ASP.NET 4.5 in C# and VB
Online Reference(s)	https://docs.microsoft.com/en-us/dotnet/core/introduction https://docs.microsoft.com/en-us/dotnet/csharp/ https://docs.microsoft.com/en-us/aspnet/core/fundamentals/choose-aspnet-framework?view=aspnetcore-5.0 https://dotnet.microsoft.com/learn/aspnet/hello-world-tutorial/intro
	https://docs.microsoft.com/en-us/learn/modules/create-razor-pages-aspnet-core/ https://docs.microsoft.com/en-us/learn/modules/build-web-api-aspnet-core/ https://youtube.com/playlist?list=PLdo4fOcmZ0oW8nviYduHq7bmKode-p8Wy https://www.youtube.com/playlist?list=PLdo4fOcmZ0oX7uTkjYwvCJDG2qhcSzwZ6

Course Code: US-SCS-407 Introduction to Research Methodology

Unit No.	Content	No. of Lectures
I	1.1 **Introduction to Research Methodology:** Meaning of Research, Objectives of Research, Motivations in Research, types of Research, Research Approaches, Significance of Research, Research Methods v/s Methodology, Research and Scientific Methods, Research Process, Criteria of Good Research. 1.2 **Defining the Research Problem:** Concept and need, Identification of Research problem, defining and delimiting Research problem. 1.3 **Formulating a Research Problem:** Reviewing Literature, formulating a Research Problem, Research Question, Identifying Variables, Constructing Hypothesis	15
II	2.1 **The Research Design**: Meaning, Need for Research Design, Important Concepts, Different Research Designs, Basic Principles of Experimental Designs. 2.2 **Tools for Data Collection**: Collections of Primary Data, Collection of Data through questionnaire and Schedules, other Observation Interview Methods, Collection of Secondary Data, Selection of appropriate method for data collection, Case Study, Focus Group Discussion, Techniques of developing research tools, viz. Questionnaire and rating scales etc. Reliability and validity of Research tools. Steps in Sampling Design, Criteria of Selecting a Sampling Procedure, Characteristics of a Good Sample Design, Different Types of Sample Designs, How to Select a Random Sample. 2.3 **Sampling:** Probability and Non-Probability sampling- types and criteria for selection. Developing sampling Frames. 2.4 **Hypothesis Testing:** What is a Hypothesis? Characteristics of good Hypothesis. Basic Concepts Concerning Testing of Hypotheses, Procedure for Hypothesis Testing, Flow Diagram for Hypothesis Testing, Tests of Hypotheses, and One sided and Two sided hypothesis, Critical region, p- value, Confidence intervals, Conducting a Hypothesis Test, Type – I and Type – II errors.Sampling Distribution, Null Hypothesis- Alternative Hypothesis. Testing the Significance of difference between means (z and t test) Analysis of Variance (ANOVA) and Analysis of covariance (ANCOVA) - concept and applications only.	15
III	3.1 **Technical Writing:** Writing a Research Proposal, what is a Scientific Paper? Ethics in Scientific Publishing. 3.2 **Preparing the Text:** How to Prepare the Title, How to List the Authors and Addresses, How to Prepare the Abstract, How to Write the Introduction, How to Write the Materials and Methods Section, How to Write the Results, How to Write the Discussion, How to State the Acknowledgments, How to Cite the References. 3.3 **Preparing the Tables and Figures:** How to Design Effective Tables, How to Prepare Effective Graphs, How to Prepare Effective Photographs. 3.4 **Publishing the Paper:** Rights and Permissions, How to Submit the Manuscript, How and When to Use Abbreviations, How to Write a Thesis, Outcome of Research	15

Tutorials:
- 8 to 10 Tutorials based on the entire syllabus.

Self-Learning topics

2.4	**Hypothesis Testing:** What is a Hypothesis? Characteristics of good Hypothesis. Basic Concepts Concerning Testing of Hypotheses, Procedure for Hypothesis Testing, Flow Diagram for Hypothesis Testing, Tests of Hypotheses, and One sided and Two sided hypothesis, Critical region, p-value, Confidence intervals, Conducting a Hypothesis Test, Type – I and Type – II errors. Sampling Distribution, Null Hypothesis- Alternative Hypothesis. Testing the Significance of difference between means (z and t test) Analysis of Variance (ANOVA) and Analysis of covariance (ANCOVA) - concept and applications only.
-	Presentation or Report Writing using LaTeX on a Case Study / Research Papers.

References: Paper-VII US-SCS-407 Introduction to Research Methodology

References	Kothari C.R., Research Methodology, New Age International Publication, New Delhi. Ranjit Kumar, Research Methodology-A Step-by-Step Guide forBeginners, (4th ed.), 2014, Singapore, Pearson Education. Robert, A. Day, How to Write and Publish a Scientific Paper, Cambridge University Press, Great Britain.
Online Reference(s)	**1.** https://nptel.ac.in/courses/109/103/109103153/ **2.** https://nptel.ac.in/courses/127/105/109105115/ **3.** https://nptel.ac.in/courses/107/108/107108011/

Part - 7 Detailed Scheme Practical

Course Code: US-SCS-4P1

Practical I	US-SCS-401 + US-SCS-402 + US-SCS-403	Total Credits: 3
Unit	**Content**	**No. of Lectures**
1, 2, 3	**US-SCS-401: Design and Analysis of Algorithms** **1.** Write Python program to perform matrix multiplication. Discuss the complexity of algorithm used. **2.** Write Python program to sort n names using Quick sort algorithm. Discuss the complexity of algorithm used. **3.** Write Python program to sort n numbers using Merge sort algorithm. Discuss the complexity of algorithm used. **4.** Write Python program for inserting an element into binary tree. **5.** Write Python program for deleting an element (assuming data is given) from binary tree. **6.** Write Python program for checking whether a given graph G has simple path from source s to destination d. Assume the graph G is represented using adjacent matrix. **7.** Write Python program for implementing Huffman Coding Algorithm. Discuss the complexity of algorithm. **8.** Write Python program for implementing Strassen's Matrix multiplication using Divide and Conquer method. Discuss the complexity of algorithm. **9.** Single Source Shortest Path Problem. **10.** Longest Common Subsequence Problem. **11.** Matrix Chain Multiplication.	45
1, 2, 3	**US-SCS-402: Advanced Java** **1.** Develop the presentation layer of Library Management software application with suitable menus. **2.** Design suitable database for Library Management System. **3.** Develop business logic layer for Library Management System. **4.** Develop Java application to store image in a database as well as retrieve image from database. **5.** Write a Java application to demonstrate servlet life cycle. **6.** Design database for student administration. Develop servlet(s) to perform CRUD operations. **7.** Create Employees table in EMP database. Perform select, insert, update, and delete operations on Employee table using JSP. **8.** Write a Student class with three properties. The useBean action declares a JavaBean for use in a JSP. Write Java application to access JavaBeans Properties. **9.** Design application using Struts2. Application must accept user name and greet user when command button is pressed. **10.** Write Java application to encoding and decoding JSON in Java.	45

1, 2, 3	**US-SCS-403: Computer Networks** **1.** Understanding the working of NIC cards, Ethernet/Fast Ethernet/Gigabit Ethernet. **2.** Crimping of Twisted-Pair Cable with RJ45connector for Straight-Through, Cross-Over, Roll-Over. **3.** To understand their respective role in networks/internet. **4.** Problem solving with IPv4, which will include concept of Classful addressing. (supportive Hint: use Cisco Binary Game) *5.* Using, linux-terminal or Windows-cmd, execute following networking commands and note the output: *ping, traceroute, netstat, arp, ipconfig.* **6.** Using **Packet Tracer**, create a basic network of two computers using appropriate network wire. **7.** Using **Packet Tracer**, connect multiple (min.6) computers using layer 2 switch. **8.** Using **Packet Tracer**, connect a network in triangular shape with three layer two switches and every switch will have four computer. Verify their connectivity with each other. **9.** Using **Packet Tracer**, create a wireless network of multiple PCs using appropriate access point. **10.** Using **Wireshark**, network analyzer, set the filter for ICMP, TCP, HTTP, UDP, FTP and perform respective protocol transactions to show/prove that the network analyzer is working.	**45**

Course Code: US-SCS-4P2

ctical II	**US-SCS-404 + US-SCS-405 + US-SCS-406**	**Total Credits: 3**
Unit	**Content**	**No. of Lectures**
1, 2, 3	**US-SCS-404: Software Testing and Quality Assurance** **1.** Install Selenium IDE; Write a test suite containing minimum 4 test cases for different formats. **2.** Conduct a test suite for any two web sites. **3.** Install Selenium server (Selenium RC) and demonstrate it using a script in Java/PHP. **4.** Write and test a program to login a specific web page. **5.** Write and test a program to update 10 student records into table into Excel file. **6.** Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects). **7.** Write and test a program to provide total number of objects present / available on the page. **8.** Write and test a program to get the number of items in a list / combo box. **9.** Write and test a program to count the number of check boxes on the page checked and unchecked count. **10.**Load Testing using JMeter, Android Application testing using Appium Tools, Bugzilla Bug tracking tools	**45**
1, 2, 3	**US-SCS-405: Linear Algebra using Python** 1. Write a program which demonstrates the following:	**45**

	- Addition of two complex numbers - Displaying the conjugate of a complex number - Plotting a set of complex numbers - Creating a new plot by rotating the given number by a degree 90, 180, 270 degrees and also by scaling by a number a=1/2, a=1/3, a=2 etc. 2. Write a program to do the following: - Enter a vector u as a n-list - Enter another vector v as a n-list - Find the vector au+bv for different values of a and b - Find the dot product of u and v 3. Write a program to do the following: - Enter two distinct faces as vectors u and v. - Find a new face as a linear combination of u and v i.e. au+bv for a and b in R. - Find the average face of the original faces. 4. Write a program to do the following: - Enter an r by c matrix M (r and c being positive integers) - Display M in matrix format - Display the rows and columns of the matrix M - Find the scalar multiplication of M for a given scalar. - Find the transpose of the matrix M. 5. Write a program to do the following: - Find the vector –matrix multiplication of a r by c matrix M with an c-vector u. - Find the matrix-matrix product of M with a c by p matrix N. 6. Write a program to enter a matrix and check if it is invertible. If the inverse exists, find the inverse. 7. Write a program to convert a matrix into its row echelon form. 8. Write a program to do the following: - Enter a positive number N and find numbers a and b such that $a_2 – b_2 = N$ - Find the gcd of two numbers using Euclid's algorithm. 9. Write a program to do the following: - Enter a vector b and find the projection of b orthogonal to a given vector u. - Find the projection of b orthogonal to a set of given vectors 10. Write a program to enter a given matrix and an eigen value of the same. Find its eigen vector.	
1, 2, 3	**US-SCS-406: .NET Technologies** **1.** Write C# programs for understanding C# basics involving **a.** Variables and Data Types **b.** Conditional Logic **c.** Loops **d.** Methods	**45**

	2. Write C# programs for Object oriented concepts of C# such as: **a.** Program using classes **b.** Constructor and Function Overloading **c.** Inheritance **d.** Namespaces **3.** Design ASP.NET Pages with **a.** Server controls. **b.** Web controls and demonstrate the use of AutoPostBack **4.** Design ASP.NET Pages for State Management using **a.** Cookies **b.** Session State **c.** Application State **5.** Perform the following activities **a.** Design ASP.NET page and perform validation using various Validation Controls **6.** Performing ADO.NET data access in ASP.NET. **7.** Design ASP.NET application to demonstrate LINQ syntax query expressions **8.** Design ASP.NET application to query a Database using LINQ **9.** Create Hello World Web App in ASP.NET Core **10.** Create an ASP.NET Core MVC web app **11.** Create a Razor Pages web app with ASP.NET Core.	

Geeta B

Ms. Geeta N. Brijwani Chairperson
BoS
Computer Science HSNC University