SHRI VENKATESHWARA UNIVERSITY

Syllabus
B.Tech

Computer Science and Engineering

V SEMESTER
(Four Years Programme)

(w.e.f. 2019-20)

SCHOOL OF ENGINEERING & TECHNOLOGY

Computer Science and Engineering

V SEMESTER

S. No.	Subject Codes	Subject	Periods			Evaluation Scheme				End Semester		Total	Credit
			L	T	P	C T	T A	Total	P S	TE	P E		
1	**SCS-501**	Database Management Systems	3	0	0	20	10	30		70		100	3
2	SCS -502	Object Oriented Programming	2	0	0	20	10	30		70		100	2
3	**SCS - 503**	Signals & Systems	3	0	0	20	10	30		70		100	3
4	**SCS-504**	Formal Language & Automata Theory	3	0	0	20	10	30		70		100	3
5	**SCS-505**	Software Engineering	3	0	0	20	10	30		70		100	3
6	**SOE-051**	Application of Psychology	2	0	0	20	10	30		70		100	2
7	**SCS -511**	Database Management Systems Lab	0	0	4				25		25	50	2
8	**SCS-512**	Object Oriented Programming Lab	**0**	**0**	**4**				25		25	50	**2**
9	**SNM-501**	Essence of Indian Traditional Knowledge	0	0	2								0
												700	**20**

SCS-501 **SCS-501**	**Database Management Systems**	**3L:0T:0P**	**3Credits**

Pre-requisites	

Objectives of the course

- To understand the different issues involved in the design and implementation of a database system.
- To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
- To understand and use data manipulation language to query, update, and manage a database
-

- To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.
- To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Detailed contents

Module 1

Database system architecture: Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).

Data models: Entity-relationship model, network model, relational and object oriented datamodels, integrity constraints, data manipulation operations.

Module 2:

Relational query languages: Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.

Relational database design: Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design.

Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

Module 3:

Storage strategies: Indices, B-trees, hashing.

Module 4:

Transaction processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.

Module 5:

Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.

Module 6:

Advanced topics: Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

Suggested books:

1. "Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.

Suggested reference books

1 "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.

2 "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education

3 "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

Course Outcomes

1. For a given query write relational algebra expressions for that query and optimize the developed expressions
2. For a given specification of the requirement design the databases using E-R method and normalization.
3. For a given specification construct the SQL queries for Open source and Commercial DBMS -MYSQL, ORACLE, and DB2.
4. For a given query optimize its execution using Query optimization algorithms
5. For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.
6. Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.

SCS-502	**Object Oriented Programming**	**2L:0T:0P**	**2 Credits**
Pre-requisites			

Objectives of the course

The course will introduce standard tools and techniques for software development, using object oriented approach, use of a version control system, an automated build process, an appropriate framework for automated unit and integration tests.

Detailed contents

- *Abstract data types and their specification.*
- *How to implement an ADT.* Concrete state space, concrete invariant, abstraction

function. Implementing operations, illustrated by the Text example.

- *Features of object-oriented programming.* Encapsulation, object identity, polymorphism – but not inheritance.
- *Inheritance in OO design.*
- *Design patterns.* Introduction and classification. The iterator pattern.
- *Model-view-controller pattern.*
- *Commands as methods* and as objects. • *Implementing OO language features.*
- *Memory management.*
- *Generic types and collections*
- *GUIs. G*raphical programming with Scala and Swing • *The software development process.*

The concepts should be practised using C++ and Java. Pearl may also be introduced wherever possible.

Suggested books

1. Barbara Liskov, *Program Development in Java*, Addison-Wesley, 2001

Suggested reference books

1. Any book on Core Java
2. Any book on C++

Course Outcomes

After taking the course, students will be able to:

1. Specify simple abstract data types and design implementations, using abstraction functions to document them.
2. Recognise features of object-oriented design such as encapsulation, polymorphism, inheritance, and composition of systems based on object identity.
3. Name and apply some common object-oriented design patterns and give examples of their use.
4. Design applications with an event-driven graphical user interface.

SCS- 503 SIGNALS AND SYSTEMS

Course objectives: This course will enable students to:

- Understand the mathematical description of continuous and discrete time signalsand systems.
- Analyze the signals in time domain using convolution difference/differentialequations
- Classify signals into different categories based on their properties.
- Analyze Linear Time Invariant (LTI) systems in time and transform domains.
- Build basics for understanding of courses such as signal processing, controlsystem and communication.

Modules	
Module -1	
Introduction and Classification of signals: Definition of signal and systems, communication and control systems as examples. Sampling of analog signals, Continuous time and discrete time signal, Classification of signals as even, odd, periodic and non-periodic, deterministic and non-deterministic, energy and power. Elementary signals/Functions: Exponential, sine, impulse, step andits properties, ramp, rectangular, triangular, signum, sync functions. Operations on signals: Amplitude scaling, addition, multiplication,differentiation, integration (Accumulator for DT), time scaling, timeshifting and time folding. Systems: Definition, Classification: linear and non-linear, time variantand invariant, causal and non- causal, static and dynamic, stable andunstable, invertible.	
Module -2	
Time domain representation of LTI System: System modeling: Input-output relation, definition of impulse response, convolution sum, convolution integral, computation of convolution integral and convolution sum using graphical method for unit step to unit step, unit step to exponential, exponential to exponential, unit step to rectangular and rectangular to rectangular only. Properties of convolution.	
Module -3	

System interconnection, system properties in terms of impulse response, step response in terms of impulse response (4 Hours). Fourier Representation of Periodic Signals : Introduction to CTFS and DTFS, definition, properties (No derivation) and basic problems (inverse Fourier series is excluded) (06 Hours).	
Module -4	
Fourier Representation of aperiodic Signals: FT representation of aperiodic CT signals - FT, definition, FT of standard CT signals, Properties and their significance (4 Hours). FT representation of aperiodic discrete signals-DTFT , definition, DTFT of standard discrete signals, Properties and their significance (4 Hours). Impulse sampling and reconstruction: Sampling theorem (only statement) and reconstruction of signals (2 Hours).	
Module -5	
Z-Transforms: Introduction, the Z-transform, properties of the Region of convergence, Properties of the Z-Transform, Inversion of the Z- Transform, Transform analysis of LTI systems.	
Course Outcomes: At the end of the course, students will be able to: • Classify the signals as continuous/discrete, periodic/aperiodic, even/odd,energy/power and deterministic/random signals. • Determine the linearity, causality, time-invariance and stability properties ofcontinuous and discrete time systems. • Compute the response of a Continuous and Discrete LTI system using convolutionintegral and convolution sum. • Determine the spectral characteristics of continuous and discrete time signal usingFourier analysis. • Compute Z-transforms, inverse Z- transforms and transfer functions of complexLTI systems.	
Question paper pattern: • The question paper will have ten questions. • Each full Question consisting of 16 marks • There will be 2 full questions (with a maximum of Three sub questions) fromeach module. • Each full question will have sub questions covering all the topics under amodule. • The students will have to answer 5 full questions, selecting one full question from each module.	
Text Book: Simon Haykins and Barry Van Veen, “Signals and Systems”, 2nd Edition,2008, WileyIndia. ISBN 9971-51-239-4. Reference Books: 1. Michael Roberts, “Fundamentals of Signals & Systems”, 2nd edition, Tata McGraw-Hill, 2010, ISBN 978-0-07-070221-9. 2. Alan V Oppenheim, Alan S, Willsky and A Hamid Nawab, “Signals and Systems” Pearson Education Asia / PHI, 2nd edition, 1997. IndianReprint 2002.	

3. H. P Hsu, R. Ranjan, "Signals and Systems", Scham's outlines, TMH, 2006.
4. B. P. Lathi, "Linear Systems and Signals", Oxford University Press, 2005.
5. Ganesh Rao and Satish Tunga, "Signals and Systems", Pearson/Sanguine Technical Publishers, 2004.

SCS-504	**Formal Language & Automata Theory**	**3L:0T:0 P**	**3 Credits**

Pre-requisites	

Objectives of the course

- Develop a formal notation for strings, languages and machines.
- Design finite automata to accept a set of strings of a language.
- Prove that a given language is regular and apply the closure properties of languages.
- Design context free grammars to generate strings from a context free language and convert them into normal forms.
- Prove equivalence of languages accepted by Push Down Automata and languages generated by context free grammars
- Identify the hierarchy of formal languages, grammars and machines.
- Distinguish between computability and non-computability and Decidability and undecidability.

Detailed contents

Module 1:

Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages. Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata. Context-free languages and pushdown automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs. Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. Turing machines: The basic model for Turing machines (TM), Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators. Undecidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.

Suggested books

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to AutomataTheory, Languages, and Computation, Pearson Education Asia.

Suggested reference books:

2. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.
3. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
4. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
5. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill.

Course Outcomes:

1. Write a formal notation for strings, languages and machines.
2. Design finite automata to accept a set of strings of a language.
3. For a given language determine whether the given language is regular or not.
4. Design context free grammars to generate strings of context free language .
5. Determine equivalence of languages accepted by Push Down Automata and languages generated by context free grammars
6. Write the hierarchy of formal languages, grammars and machines.
7. Distinguish between computability and non-computability and Decidability and undecidability.

Course Code	:	SCS -505
Course Title	:	Software Engineering
Number of Credits	:	3 (L: 3, T: 0, P: 0)
Prerequisites	:	-
Course Category	:	PC

Course Learning Objectives:

Inculcate essential technology and software engineering knowledge and skills essential to build a reasonably complex usable and maintainable software iteratively. 2) Emphasize on structured ap- proach to handle software development. 3) Enhance communication skills.

Course Content: As per the course design, concepts learned as part of this course will/should be used in the Minor Project (Proj.202). These two courses should go hand in hand to be effective.

UNIT 1:

Introduction to Software Engineering, Lifecycle, Process Models - Traditional v/s Agile processes.

UNIT 2:

Development Activities - Requirements Gathering and Analysis, Design Concepts, Software architec- ture and Architectural styles, Basic UI design, Effective Coding and Debugging techniques.

UNIT 3:

Software Testing Basics, Unit, Integration, System and Acceptance Testing, Introduction to various testing techniques (e.g. Stress testing), Writing and executing test cases, Quality Assurance.

UNIT 4:

Project Management - Project management concepts, Configuration and Release Management, Ver- sion Control and its tools (Git), Release Planning, Change Management, Software Maintenance, Proj-ect Metrics.

Reference Books:

1. Software Engineering – A Practitioner's Approach, 7th Edition, Roger Pressman.
2. Software engineering, Ian Sommerville, Pearson Education
3. An Integrated Approach to Software Engineering, Pankaj Jalote, Springer Verlag
4. Software Engineering, Nasib Singh Gill, Khanna Book Publishing Co. India.
5. Software Engineering, K. K. Agarval, Yogesh Singh, New Age International Publishers

Course outcomes:

The proposed course is expected to provide an introduction to software engineering concepts and techniques to undergraduate students, thus enabling them to work in a small team to deliver a soft-ware system. The course content and project will introduce various software technologies, process and project management skills that are needed for the delivery of software in a team setting.

As per the course design, concepts learned as part of this course will/should be used in the Minor Project (Proj.202). These two courses should go hand in hand to be effective.

UNIT 1:

Introduction to Software Engineering, Lifecycle, Process Models - Traditional v/s Agile processes.

UNIT 2:

Development Activities - Requirements Gathering and Analysis, Design Concepts, Software architec- ture and Architectural styles, Basic UI design, Effective Coding and Debugging techniques.

UNIT 3:

Software Testing Basics, Unit, Integration, System and Acceptance Testing, Introduction to various testing techniques (e.g. Stress testing), Writing and executing test cases, Quality Assurance.

UNIT 4:

Project Management - Project management concepts, Configuration and Release Management, Ver- sion Control and its tools (Git), Release Planning, Change Management, Software Maintenance, Proj-ect Metrics.

Reference Books:

1. Software Engineering – A Practitioner's Approach, 7th Edition, Roger Pressman.
2. Software engineering, Ian Sommerville, Pearson Education
3. An Integrated Approach to Software Engineering, Pankaj Jalote, Springer Verlag
4. Software Engineering, Nasib Singh Gill, Khanna Book Publishing Co. India.
5. Software Engineering, K. K. Agarval, Yogesh Singh, New Age International Publishers

Course outcomes:

The proposed course is expected to provide an introduction to software engineering concepts and techniques to undergraduate students, thus enabling them to work in a small team to deliver a soft- ware system. The course content and project will introduce various software technologies, process and project management skills that are needed for the delivery of software in a team setting.

APPLICATIONS OF PSYCHOLOGY
Course code: SOE-051

Credits: 2-0-0

1. OBJECTIVES:

The objectives of this course are to make students:

1) aware of the different applications of psychology to everyday issues of life,
2) aware of the different social issues, workplace issues, and behavioural issues, and
3) understand how the knowledge gained from this course can be used in their own personal and professional work life.

2. COURSE TOPICS:

2.1 Unit 1: Introduction: Nature and fields. (6)

2.2 Unit 2: Psychology in industries and organizations: Job analysis; fatigue and accidents; consumer behavior. (8)

2.3 Unit 3: Psychology and mental health: Abnormality, symptoms and causes psychological

disorders.(10)

2.4 Unit 4: Psychology and Counseling: Need of Counseling, Counselor and the Counselee, Counseling Process, Areas of Counseling. (6)

2.5 Unit 5: Psychology and social behavior: Group, group dynamics, teambuilding, Prejudice and stereotypes; Effective Communication, conflict and negotiation. (10)

Text

1. Schultz, D. & Schultz, S.E. (2009). Psychology and Work Today (10th ed.). New Jersey:Pearson/Prentice Hall.
2. Butcher, J. N., Mineka, S., & Hooley, J. M. (2010). Abnormal psychology (14th ed.). NewYork: Pearson
3. Gladding, S. T. (2014). Counselling: A comprehensive profession. New Delhi: PearsonEducation
4. Aronson, E., Wilson, T. D., & Akert, R. M. (2010). Social Psychology (7th Ed.). UpperSaddle River, NJ: Prentice Hall.

SNM-501 Essence of Indian Knowledge Tradition

□□□□□□□□Ę□□□□ **- 1**

Course objective: The course aims at imparting basic principles of thought process, reasoning and inferencing. Sustainability is at the core of Indian Traditional knowledge Systems connecting society and nature. Holistic life style of yogic science and wisdom capsules in Sanskrit literature are also important in modern society with rapid technological advancements and societal disruptions.

Part-I focuses on introduction to Indian Knowledge Systems, Indian perspective of modern scientific world-view, and basic principles of Yoga and holistic health care system.

Course Contents – Basic structure of Indian Knowledge System: अ'□□□□□Ę□ -□□□□,□□□□□□ (□□□□ő द, □□□□ő द, ग ृवDŋ□□□,'□□□ŕ □□□) □□□□□□□ (□□Ħ□, क ब, □□Ŝb, fŋ□□□□, Ù□□□□, □□□) ४ □□□□□□ (ध मDŋ□□ ‹, □□□□□□□, □□□□□, त कDŋ□□ ‹) – Modern Science and Indian Knowledge System – Yoga and Holistic Health care – Case studies

Philosophical Tradition (सवDŋदशDŋन)-Ɋ□□, □□□□□□□, □□□ƥ, □□□, □□□□□□□, □□□□□□, □□□□Dŋक, □□□, □□Ŋ – Indian Linguistic Tradition (Phonology, morphology, syntax and semantics) – Indian Artistic Tradition – चित्रकला, मूर्तिकला साहित्य Case studies

References

• V. Sivaramakrishnan (Ed.), Cultural Heritage of India-course material, Bharatiya Vidya Bhavan, Mumbai. 5th Edition, 2014
• Swami Jitatmanand, Modern Physics and Vedant, Bharatiya Vidya Bhavan • Swami Jitatmanand, Holistic Science and Vedant, Bharatiya Vidya Bhavan
• Fritzof Capra, Tao of Physics
• Fritzof Capra, The Wave of life
• VN Jha (Eng. Trans.), Tarkasangraha of Annam Bhatta, International Chinmay Foundation, Velliarnad, Arnakulam
• Yoga Sutra of Patanjali, Ramakrishna Mission, Kolkata
• GN Jha (Eng. Trans.), Ed. RN Jha, Yoga-darshanam with Vyasa Bhashya, Vidyanidhi Prakashan, Delhi 2016
• RN Jha, Science of Consciousness Psychotherapyand Yoga Practices, Vidyanidhi Prakashan, Delhi 2016 • P B Sharma (English translation), Shodashang Hridayan

Pedagogy: Problem based learning, group discussions, collaborative mini projects.
Outcome: Ability to understand, connect up and explain basics of Indian traditional knowledge in modern scientific perspective.