Rayat Shikshan Sanstha’s

Karmaveer Bhaurao Patil Mahavidyalya, Pandharpur Dist-Solapur(MH)

Autonomous

NAAC Reaccredited “A$^+$” grade, CGPA: 3.51

Granted under FIST-DST and The Best College

Affiliated to the

Punyashlok Ahilyadevi Holkar, Solapur University, Solpaur.

Program : B.Sc(ECS)-II

Semester: III and IV

(Choice Based Credit, Grading and Semester System with effect from the academic year 2020-2021)

Rayat Shikshan Samnstha's

Karmaveer Bhaurao Patil Mahavidyalya,Pandharpur
(Autonomous)
Syllabus for Bachelor of Science Part – II

- **TITLE** : B.Sc. (Entire Computer Science)
- **YEAR OF IMPLEMENTATION** : 2020-21

PREAMBLE:

There are bright career prospects for computer science professionals or software professionals in recent scenario. With the opening of huge software and IT companies in India, the job opportunities for trained professionals have increased considerably. India is known to be a leader in software and IT sector.

Computer science graduates pass outs find job opportunities in a variety of environments in academia, research, industry, government, private, business organizations and so on.

They are involved in analyzing problems for solutions, formulating and testing, using advanced communications or multi-media equipment, or working in teams for product development.

The software and IT companies are the major employers of computer science graduates. They offer the best packages to the young graduates which are unmatched with other branches of science.

- **GENERAL OBJECTIVES OF THE COURSE :**
  - The content of the syllabus have been framed as per UGC norms of CBCS Pattern.
  - The students are expected to understand the fundamentals, principles, mathematical, recent IT concepts and recent developments in the subject area.
  - The practical course is in relevance to the theory courses to improve the understanding of the concepts.
  - It is expected to inspire and boost interest of the students towards Computer Science as the main subject.
  - To develop the power of appreciations, the achievements in Computer and role in nature and society.
  - To enhance student sense of enthusiasm towards IT and to involve them in an intellectually stimulating experience of learning in a supportive environment.
- **DURATION** : 3 YEAR
- **PATERN** : CBCS SEMISTER
- **MEDIUM OF INSTRUCTION** : ENGLISH
- **STRUCTUTRE OF COURSE** :

Rayat Shikshan Sanstha's

Karmaveer Bhaurao Patil Mahavidyalaya,Pandharpur

(Autonomous)

Affiliated to Punyashlok Ahilyadevi Holkar Solapur University,Solapur

Faculty of Science Choice Based Credit System (CBCS)

Structure for B.Sc(Entire Computer Science) -II ,Sem-III,IV(w.e.f June 2020-21)

Sr.No.	Subject Code	Subject Name	Hours/Week			CI	SEE	Total Marks per Paper	Credit
			L	T	P				
Semester - III									
1	KBP-S-ECS-2319	Data Structure using C++ -I	3			10	40	50	4
2	KBP-S-ECS-2320	Data Structure using C++ -II	3	--	--	10	40	50	
3	KBP-S-ECS-2321	Software Engineering	3	--	--	10	40	50	4
4	KBP-S-ECS-2322	Software Testing	3	--	--	10	40	50	
5	KBP-S-ECS-2323	Probability Theory –I	3	--	--	10	40	50	4
6	KBP-S-ECS-2324	Statistics for Data Science	3	--	--	10	40	50	
7	KBP-S-ECS-2325	Introduction to Python programming	3	--	--	10	40	50	2
8	KBP-S-ECS-AECC-2326-COMP	Democracy Election and Good Governances	3	--	--	10	40	50	NC
		Total(Theory)	**24**	**--**	**--**	**80**	**320**	**400**	**14**

Rayat Shikshan Sanstha's

Karmaveer Bhaurao Patil Mahavidyalaya,Pandharpur

(Autonomous)

Affiliated to Punyashlok Ahilyadevi Holkar Solapur University,Solapur

Faculty of Science Choice Based Credit System (CBCS)

Structure for B.Sc(Entire Computer Science) -II ,Sem-III,IV(w.e.f June 2020-21)

Sr.No.	Subject Code	Subject Name	Hours/Week			CI	SEE	Total Marks Per Paper	Credit
			L	T	P				
Semester - IV									
1	KBP-S-ECS-2427	Database Management System	3			10	40	50	4
2	KBP-S-ECS-2428	Mysql Server	3	--	--	10	40	50	
3	KBP-S-ECS-2429	Operating System	3	--	--	10	40	50	4
4	KBP-S-ECS-2430	Linux OS and Shell Scripting	3	--	--	10	40	50	
5	KBP-S-ECS-2431	Probability Theory-II	3	--	--	10	40	50	4
6	KBP-S-ECS-2432	Optimization Techniques	3	--	--	10	40	50	
7	KBP-S-ECS-2433	Programming with Python	3	--	--	10	40	50	2
8	KBP-S-ECS-2434	Environmental studies	3	--	--	10	40	50	NC
		Total(Theory)	**24**	**--**	**--**	**80**	**320**	**400**	**14**
9	KBP-S-ECS-Lab-P-VI	Practical Based on (231,232,241,242,237)	--	--	8	20	80	100	4
10	KBP-S-ECS- Lab-P-VII	Practical Based on (233,234,243,244,247)	--	--	8	20	80	100	4
11	KBP-S-ECS- Lab-P-VIII	Practical Based on (235,236,245,246)	--	--	8	20	80	100	4
		Total(Practical's)	**--**	**--**	**24**	**60**	**240**	**300**	**12**
		Grant Total	**48**	**--**	**24**	**220**	**880**	**1100**	**40**

B.Sc(ECS)-II Semester-III

Data Structure using C++ **Theory Lectures 30**

Learning objectives:

1. To bring out the importance of data structures in a variety of applications.
2. To introduce linear (arrays, linked list, doubly linked list) and non linear data structures (Binary Tree, Heap).
3. To present the advantages and applications of hashing.

KBP-S-ECS-231 Data Structure using C++ - I

Unit - I **(07Lectures)**

Introduction: Need of Data Structure, Types of Data Structure, ADT, Algorithm: Definition, characteristics, Space complexity, time complexity, Asymptotic notation (Big O, Omega Ώ, theta Φ)

Unit - II **(14Lectures)**

Stack: Introduction to stack, Representation-static & dynamic, stack Operations, Application -infix to postfix & prefix, postfix evaluation, recursion, expression validity.

Queues: Introduction to Queue, Representation -static & dynamic, Operations, Circular queue, De-queue, priority queues.

Unit - III **(09Lectures)**

Linked List:-Introduction to List, Implementation of List – static & dynamic representation, Types of Linked List, Operations on List, Applications of Linked List – polynomial manipulation

KBP-S-ECS-232 Data Structure using C++ - II **Theory Lectures 30**

Unit - I **(09Lectures)**

Trees: Concept & Terminologies, Binary tree, binary search tree, Representation – static &dynamic, Operations on BST – create, Insert, delete, traversals (preorder, inorder, postorder), counting leaf, non-leaf & total nodes,

Unit - II **(07Lectures)**

Algorithm design strategies: Divide and Conquer, Greedy, Dynamic programming, Backtracking, Branch and Bound.

Unit - III **(14Lectures)**

Sorting: Bubble sort, Quick sort, Simple Insertion sort, Shell sort, Address calculation sort, Binary Search Tree, Heap Sort, Merge sort, Radix Sort.

Searching: Linear Search, Binary Search, Tree searching methods, Multiway search tree (B-tree, B+ tree), Height balance tree- AVL trees-Rotations. Hash function(open and close)

Reference Books

1 . Aho, Hopcroft, Ulman: Data structures and Algorithms.

2. Nikaulus Wirth: Algorithms, data structures, Programs.
3. ThomsHorbron: File Systems, Structures and Algorithms (PHI).
4. D. E. Kunth: Art of computer Programming Vol – I.
5. Tanenbaum: Data structures using C and C++ (PHI).
6. fundamentals of computer algorithms by ellis horowitz sartaj sahni 2nd edition galgotia publication

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

1. Identify the appropriate and optimal data structure for a specified application.
2. Employ the benefits of dynamic and static data structures implementations
3. Illustrate the use of different non-linear data structures and their applications.
4. Demonstrate the use of techniques like hashing, trees and heaps in a variety of applications.

KBP-S-ECS-233 Software Engineering **Theory Lectures 30**

Learning Outcomes:

1 Software Engineering Students will try to learn:
The main objective of the course is to introduce to the students about the product that is to be engineered and the processes that provides a framework for the engineering methodologies and practices.
2. To provide the knowledge of software engineering discipline.
3. To apply analysis, design and testing principles to software project development.
4. To demonstrate and evaluate real time projects with respect to software engineering principles.

Unit - I : Introduction To Software Engineering and Process Models (7 Lectures)

Definition of Software,Nature of Software Engineering,Changing nature of software, Software Process,The Process Framework,Umbrella Activities, Process Adaptatio, Generic Process Mode, Prescriptive Process Models, The Waterfall Model, Incremental Process Models,Evolutionary Process Models, Concurrent Models, The Unified Process

Unit - II: Agile Development (5 Lectures)

What is Agility? Agile Process,Agility Principles,The Politics Of Agile Development, Human Factors,Extreme Programming(XP),XP Values,XP Process,Industrial XP,Adaptive Software Development(ASD),Scrum, Dynamic System Development Model (DSDM),Agile Unified Process (AUP)

Unit - III : Requirements Analysis (6 Lectures)

Requirement Elicitation, Software requirement specification (SRS),Developing Use Cases (UML),Building the Analysis Model,Elements of the Analysis Model, Analysis Patterns,Agile Requirements Engineering, Negotiating Requirements ,Validating Requirements

Unit -IV : Requirements Modeling (6 Lectures)

Introduction to UML,Structural Modeling,Use case model,Class model, Behavioral Modeling,Sequence model, Activity model,Communication or Collaboration model,Architectural Modeling,Component model,Artifact model ,Deployment model.

Unit - V : Design Concepts (6 Lectures)

Design Process,Software Quality Guidelines and Attributes, Evolution of Software Design,Design Concepts, Abstraction, Architecture,Patterns, Separation of Concerns,Modularity, Information Hiding, Functional Independence, Refinement,Aspects,Refactoring, Object Oriented Design Concepts, Design Classes,Dependency Inversion,Design for Test,The Design Model,Data Design Elements,Architectural Design Elements, Interface Design Elements, Component-Level Diagram,Deployment-Level Diagram.

Reference Books:

1. Software Engineering : A Practitioner’s Approach - Roger S. Pressman, McGraw hill(Eighth Edition) ISBN-13: 978-0-07-802212-8, ISBN-10: 0-07-802212-6

2. A Concise Introduction to Software Engineering - Pankaj Jalote, Springer ISBN: 978-1-84800-301-9

3. The Unified Modeling Language Reference Manual - James Rambaugh, Ivar Jacobson, Grady Booch ISBN 0-201-30998-X

Learning Outcomes :

Students will be able to:
1. Understand and demonstrate basic knowledge in software engineering.
2. Identify requirements, analyze and prepare models.
3. Plan, schedule and track the progress of the projects.
4. Design & develop the software projects.
5. Identify risks, manage the change to assure quality in software projects.

KBP-S-ECS-234 Software Testing **Theory Lectures 30**

Learning Objectives :

1. To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
2. To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
3. To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
4. To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.
5. To understand software test automation problems and solutions.
6. To learn how to write software testing documents, and communicate with engineers in various forms.

Unit - I **(08 Lectures)**

Introduction To Software Testing: What is Software Testing? ,Use or need of software testing.

Software Development Life Cycle (SDLC) : Water Fall Model,Spiral Model, V- Model , Prototype Model ,Hybrid Model

Unit Testing:

Unit - II **(12 Lectures)**

Software Test Automation : What is Test Automation, Terms used in automation, Skills needed for automation, What to automate, scope of automation, Design and Architecture of automation, Generic requirement for Test Tool, Process Model for Automation, Selecting Test Tool, Automation for XP/Agile model, Challenges in Automation, Data-driven Testing. Automation Tools like JUnit, Jmeter

Unit - III **(10Lectures)**

Test cases and its design Techniques: Introduction to Test Case , Characteristics Of Good Test Case,Test Case Template,How To Write A Test Case,How To Ensure The Test Coverage Is Good , How To Identify whether It Is a Good Test Case Or Not Review Process/Peer Review , Preparing Review Report ,Examples On Writing Test Cases ,Test Cases Design Techniques-Error Guessing,Equivalence Partitioning, Boundary Value Analysis.

Software Test Life cycle and Defect Life Cycle:

Software Test Life Cycle- Writing Test Plan,Preparing Traceability Matrix,Writing Test Execution Report,Summary Report,Retrospect Meeting / Triage Meetings.

Defect Life Cycle-Concept of Defect life cycle Difference between Bug, Defect, Failure, Error.

Reference Books :

1) The art of Software Testing– Glenford J. Myers
2) Lessons learned in Software Testing –CemKaner, James Bach, Bret Pettichord
3) A Practitioner's Guide to Software Test Design- Lee Copeland
4) Software Testing Techniques, 2nd edition- Boris Beizer
5) How to Break Software: A Practical Guide to Testing- James Whittake

Learning Outcomes :

Students will be able to:

1. Have an ability to apply software testing knowledge and engineering methods.
2. Have an ability to design and conduct a software test process for a software testing project.
3. Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.
4. Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.
5. Have an ability to use various communication methods and skills to communicate with their teammates to conduct their practice-oriented software testing projects.
6. Have basic understanding and knowledge of contemporary issues in software testing, such as component-based software testing problems

KBP-S-ECS-235 Probability Theory – I **Theory Lectures 30**

Learning Objectives :

1) Define event, outcome, trial, simple event, sample space and calculate the probability that an event will occur.
2) Calculate the probability of events for more complex outcomes.
3) Solve applications involving probabilities.

Unit - I **(08 Lectures)**

Permutations & Combinations: Principles of counting, Permutations of ' n ' dissimilar objects taken ' r ' at a time (with without repetitions), Permutations of n objects not all of which r different, Combinations of n objects taken r at a time, Combinations with restriction on selection (excluding or including a particular object in the group), Numerical problems.

Unit - II **(07 Lectures)**

Probability: Idea of deterministic & nondeterministic models, Random expt. – Sample space (finite, infinite, countable), Events-Types of events, Probability – Classical def., relative frequency approach, probability models, axioms of probability, probability of an event, Theorems of probability (with proof)- i) $0\leq P(A) \leq 1$, ii) $P(A) + P(A') = 1$, iii) $P(\Phi) = 0$ iv) $P(A) \leq P(B)$ when A is subset of B v) Addition law of probability. Concept & def. of conditional probability, multiplication theorem: Concept & def. of conditional probability, multiplication theorem, Concept & def. of independence of two events, pair wise & complete independence with resp. to three events, Numerical problems.

Unit - III **(15 Lectures)**

Discrete random variable: Def. of r.v., discrete r.v., Def. of p.m.f., c.d.f. & properties of c.d.f., Def. of expectation & variance, theorems on expectation, Determination of median & mode using p.m.f., Numerical problems. Standard Discrete Distribution: Uniform Distribution- Def., mean, variance, illustration of real life situations, Binomial distribution- Def., mean, variance, illustration of real life situations, additive property (statement only). Poisson distribution- mean, variance, illustration of real life situation, additive property (Statement only), Limiting case of binomial distribution (Statement only), Hypergeometric distribution – mean, variance, illustration of real life situation, Numerical problems. 14 / 27.

Reference Books :

1. Fundamentals of Mathematical Statistics- Kapoor & Gupta.
2. Modern elementary Statistics – J.E.Freund
3. Statistical Methods – J.Medhi.
4. Fundamentals of Statistics-S.C.Gupta.
5. Fundamentals of applied Statistics-Gupta & Kapoor.
6. Business Statistics – S. Shah
7. Programmed Statistics - B.L.Agarwal.

Learning outcomes

Students will be able to:

1. Calculate probabilities by applying probability laws and theoretical results.
2. Identify an appropriate probability distribution for a given discrete or continuous random variable and use its properties to calculate probabilities.
3. Derive probability distributions of functions of random variables.
4. Derive expressions for measures such as the mean and variance of common probability distributions using calculus and algebra.
5. Calculate probabilities for joint distributions including marginal and conditional probabilities.
6. For jointly distributed random variables calculate their covariance and correlation and determine whether they are independent.
7. Apply results from large-sample theory and the Central Limit Theorem to approximate a sampling distribution.
8. Implement basic simulation methods using statistical software to investigate sampling distributions.

KBP-S-ECS-236 Statistics for Data Science **Theory Lectures 30**

Learning Objectives :

1. Students will develop relevant programming abilities.
2. Students will demonstrate proficiency with statistical analysis of data.
3. Students will develop the ability to build and assess data-based models.
4. Students will execute statistical analyses with professional statistical software.
5. Students will demonstrate skill in data management.
6. Students will apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively

Unit - I **(14 Lectures)**

1) Graphical and Diagrammatic Representation – Histogram(equal and unequal width),Ogive curves,Pie diagram.
2) Measure of central tendency – Concept of central tendency , A.M. – Definition, calculation for grouped and ungrouped data, merits and demerits. Mode - Definition, calculation for grouped and ungrouped data, merits and demerits, graphical presentation.

   Median - Definition, calculation for grouped and ungrouped data, merits and demerits,graphical presentation,introduction to quartiles.Numerical examples.
3) Measures of Dispersion – Concept of dispersion,Absolute and relative measure of dispersion,Variance,S.D. and C.V.,numerical examples.

Unit – II Correlation and Regression **(08 Lectures)**

Correlation – Concept of bivariate data, concept of correlation,scatter diagram,Karl Pearson's correlation coefficient,Rank correlation,interpretation.

Regression - Concept of regression, Regression lines,Regression coefficients,properties of regression coefficients.numerical examles

Unit - III Multiple Rgression,multiple and partial correlation. **(08 Lectures)**

Concept of multiple regression, Yule's notation,Planes of regression,partial regression coefficients.Multiple correlation coefficient – Definition and formula, Partial correlation coefficient - Definition and formula, numerical examples

Reference Books:

1. Fundamentals of mathematical statistics – Gupta and Kapoor
2. Modern elementary statistics – J.E. Fruend
3. Statistical Methods – j. Medhi
4. Programmed Statistics – B.L.Agarwal
5. Fundamental of Statistics – S.C.Gupta.

Learning Outcomes :

Students will be able to:

1. Use concepts and methods of mathematical disciplines relevant to data analytics and statistical modeling
2. Utilize statistical concepts of data analysis, data collection, modeling, and inference
3. Employ algorithmic problem-solving skills to the problem at hand, including defining clear requirements to a problem, decomposing the problem, using efficient strategies to arrive at an algorithmic solution, and implementing solutions through programming in a suitable high-level language

KBP-S-ECS-237 Introduction to Python programming

Theory Lectures 30

Learning objectives :

1. To acquire programming skills in core Python.
2. To acquire Object Oriented Skills in Python
3. To develop the skill of designing Graphical user Interfaces in Python
4. To develop the ability to write database applications in Python

Unit - I **(12Lectures)**

Introduction to Python Programming: Features/characteristic of Python, Basic syntax,Writing and executing simple program,Basic Data Types,Declaring variables,Performing assignments, arithmetic operations, Simple input-output,Precedence of operators,Type conversion.
Conditional Statements: if, if-else, nested if –else.
Looping: for, while, nested loops, Terminating loops, skipping specific conditions.

Unit - II **(10Lectures)**

String, collection lists and Tuples: Declaring strings, String Manipulation using string functions.
Introduction to Collection lists:Introduction to Collection list,Manipulating Collections Lists.
Tuples- Introduction to Tuples, Manipulating Tuples.
Dictionaries , Functions and Modules: Concept of dictionary ,Techniques to create, update &delete dictionary items.
Functions: Defining a functionCalling a function ,Advantages of functions ,Types of functions,Function parameters , Formal parameters,Actual parameters,Anonymous functions ,Global and Local variables.**Modules:**Importing module,Creating & exploring modules Mathmodule, Random module, Time module.

Unit - III **(08Lectures)**

Object Oriented Programming: Features, Concept of Class & Objects, Constructor, Types of Variables, Namespaces, Types of Methods, Inner Classes, Constructors in Inheritance, Overriding Super Class Constructors and Methods, Types of Inheritance, Abstract Classes and Interfaces, The Super( ) Method, Operator Overloading, Method Overloading, Method Overriding

Reference Books :

1) Introduction to Computer Science using Python- Charles Dierbach
2) Beginning Python: Using Python 2.6 and Python 3- James Payne
3) Practical Programming: An Introduction toComputer Science Using Python 3- Paul Gries , Jennifer Campbell, Jason Montojo
4) Programming Languages – Principles and Paradigms- AdeshPandey
5) MySQL for Python: Database Access Made Easy- A. Lukaszewski

Learning Outcomes :

Student will be able to

1. Explain basic principles of Python programming language

2. Implement object oriented concepts,

3. Implement database and GUI applications.

B.Sc(ECS)-II Semester-IV

KBP-S-ECS-241 Database Management System **Theory Lectures 30**

Learning objectives :

1. To discuss and realize the importance of Database Architecture Design notations, ER Modeling, Mapping and Schema design.
2. To gain the knowledge Relational algebra.
3. To introduce formal database design approach through normalization and discuss various normal forms.
4. To understand the importance of Concurrent Transactions and discuss issues and transaction control algorithms.

Unit - I **(05Lectures)**

Introduction to database system:-Definition, Limitations of traditional file system, indexing, Advantages of DBMS, Components of DBMS, Database Architecture, Database Users, Schemas and instances, 2 tier and 3 tier architecture, Database languages, Types of data models- relational, Network, Hierarchical, Distributed, hybrid.

Unit - II **(09Lectures)**

Relational Model and Database Design:-Relation, Domain, Tuples, types of keys, relational integrity rules, Codd's rules,

Relational Algebra operations:- Select, Project, Cartesian Product, Union, Set difference, Natural Join, Outer Join, lossless joins.

Unit - III **(16Lectures)**

Transaction Management & Concurrency Control: -Introduction, properties, transaction states, scheduling, conflict and view serializability, Introduction to Concurrency Control, problems of concurrency control., log based protocols, timestamp based protocol, deadlock, deadlock handling.

Database recovery and Atomicity:-Introduction, recovery algorithms, log base recovery, shadow paging, recovery with concurrent transaction, checkpoints or sync points or save points.

Reference Books:

1. Database System Concepts By KorthSilberschetz
2. Fundamentals of Database Systems by Elmsari, Navathe
3. Teach Yourself SQL in 14 Days by Jeff Parkins and Bryan Morgan
4. An Introduction to Database Systems by Bipin Desai

Learning Outcomes :

1. Apply the database concepts and design database for given information system.
2. Apply the concepts of Normalization and design database which possess no
3. Explain the issues of transaction like concurrency control, recovery and security.

KBP-S-ECS-242 Mysql Server **Theory Lectures 30**

Learning Objectives :

1.Transform business requirements into an operational database.
2.Create physical relational database tables to implement a database design.
3.Create, maintain, and manipulate database objects.

Unit - I **(08Lectures)**

Introduction to Mysql Server: History, Generations and characteristics, difference between DBMS & RDBMS. SQL: Introduction to SQL, Features of SQL, Basic data types, SQL statements, Set operations in SQL, Nested queries, GRANT and REVOKE, Commit, Rollback, Savepoint. SQL functions: MAX, MIN SORT, COUNT, AVERAGE, Numeric, String, Date Functions, Type conversion functions.

Unit - II **(08Lectures)**

Table – Constraints Definitions, Select with operators like arithmetic, comparison and logical, Order by and Group by clause.
Join concept: Simple, Equi, non-equi, Self, Outer join.
View- Introduction, Create, Update, Drop, Index.

Unit - III **(07Lectures)**

PL/SQL: Introduction, Difference between SQL AND PL/SQL, Block definition structure and Data types, Block Functions - %Type, %RowType, Control statements, Looping statements and sequential statement.
Packages in PL/SQL
Exception Handling in PL/SQL

Unit - IV **(07Lectures)**

Procedures-Definition, creating procedures, passing parameters.
Function-Definition, syntax and calling methods, passing parameters.
Cursors– Definition, syntax and types of cursor.
Triggers – Definition, Syntax and parts of triggers, types of triggers, enabling and disabling triggers.

Reference Books:

1) High Performance MySQL: Optimization, Backups, Replication, and More, by Baron Schwartz, Peter Zaitsev, Vadim Tkachenko, Jeremy Zawodny, Arjen Lentz, Derek J. Balling.
2) MySQL (4th Edition), by Paul DuBois
3) MySQL Stored Procedure Programming, by Guy Harrison, Steven Feuerstein.

Learning Outcomes :

Student will be able to

1. Preparation of background materials and documentation needed for Technical Support using SQL and PL/SQL.
2. Use the Relational model and how it is supported by SQL and PL/SQL.
3. Use the PL/SQL code constructs of IF-THEN-ELSE and LOOP types as well as syntax and command functions.
4. Solve Database problems using Oracle 11g SQL and PL/SQL. This will include the use of Procedures, Functions, Packages, and Triggers.

KBP-S-ECS-243 Operating System **Theory Lectures 30**

Learning Objectives :

1. To learn the fundamentals of Operating Systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication
3. To learn the mechanisms involved in memory management in contemporary OS
4. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
5. To know the components and management aspects of concurrency management
6. To learn programmatically to implement simple OS mechanisms

Unit - I **(04Lectures)**

Introduction Operating System:- Definition Operating systems, Types of Operating Systems-Batch, Multiprogramming, Time-Sharing, Real-Time, Distributed, Parallel., OS Service, System components, System Calls.

Unit - II **(16Lectures)**

Process Management:-Concept of Process, Process states, Process Control Block, Context switching, Operations on Process, Co-operating Process, Threads – Types of threads, Benefits of threads.

Concept of Process Scheduling- Types of Schedulers ,Scheduling criteria , Scheduling algorithms : Preemptive and Non-pre emptive , FCFS, SJF, Round Robin, Priority Scheduling, Multilevel Queue Scheduling, Multilevel- feedback Queue Scheduling.

Process Synchronization and Deadlocks:-The Producer Consumer Problem, Race Conditions, Critical Section Problem, Semaphores, and Classical Problems of Synchronization: Reader-Writer Problem, Dinning Philosopher Problem, Critical Regions.Definition, System Model, Dead Lock Characterization, Resource Allocation Graph, Methods of Handling Dead Locks- Deadlock Prevention, Deadlock Avoidance -banker’s lgorithm,resource-request algorithm, Deadlock detection and Recovery.

Unit - III **(10Lectures)**

Memory Management:-Basic Hardware Address Binding, Logical and Physical address Space, Dynamic Loading, Overlays, Swapping,

Memory allocation: Contiguous Memory allocation – Fixed and variable partition – Internal and External fragmentation and Compaction, Paging, Segmentation. Basics of Virtual Memory, demand paging, Page fault, Page Replacement policies: Optimal (OPT), First in First Out (FIFO), Least Recently used (LRU), Thrashing.

Storage Management:- File Management: File concept, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping).

Reference Books:

1. System programming and O.S. By D.M. Dhamdhere.
2. Modern O.S. By Andrews Tanenbaum.
3. Operating System Concepts By Siberchatz and Galvin.

Learning Outcomes :

Students will be able to:

1. Analyze the structure of OS and basic architectural components involved in OS design.
2. Analyze and design the applications to run in parallel either using process or thread models of different OS.
3. Analyze the various device and resource management techniques for timesharing and distributed systems.
4. Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system.
5. Interpret the mechanisms adopted for file sharing in distributed Applications
6. Conceptualize the components involved in designing a contemporary OS

KBP-S-ECS-244 Linux OS and Shell Scripting **Theory Lectures 30**

Learning Objectivise :

1. To provide introduction to Linux operating system and its File System.
2. To develop the ability to formulate Regular Expressions and use them for Pattern Matching.
3. To gain an understanding of important aspects related to the Shell and the Process.
4. To provide a comprehensive introduction to Shell Programming, Services and Utilities.

Unit - I **(08Lectures)**

Introduction of Linux:-History of Linux, Architecture of Linux system & features, Kernel, Shell & its type, Difference between Windows and Linux.
Linux Distributions, Working environments: KDE, GNOME, Xface4, Hardware requirement,Installation procedure of Linux, Create partitions, Configuration of X system Users & Groups Management:- Create Users, Create groups, Specialgroups, Assigning permissions to users and Groups, File and Directorypermissions chmod, chown, chgrp.
Linux File System:-Hierarchy of File system,File System parts Boot Block, Super Block, Inode Block, Data Block, File types, Devices and Drives in Linux, Mounting devices (CD/DVD, usb, hard drive partition), file system

Unit - II **(12Lectures)**

Linux commandsFile and directory Management Commands:-mkdir, rmdir, cd and pwd, file, ls, cat, more, less, File and Directory Operations: find, cp, mv, rm, lnetc, Printing the files lpr, lpq, lprm etc.
Filter Commands & Editor:-Filters: head, tail , pr, cut, paste, sort, uniq, tr, grep, egrep, fgrep, sed.
Communication commands:-mesg, talk, write, wall, mail.
Text Editors: vi, vim Archive and File compression commands
Shell Programming:- Shell Variables, Metacharacters, Shell Scripts – Control and Loop structure, I/O and Redirection, Piping,

Unit - III **(10Lectures)**

Process Management: Shell process, Parent and children, Process status, System process, Multiple jobs in background and foreground, Changing process priority with nice. Listing processes, ps, kill, premature termination of process.
Disk management and System Administration:-Disk Partitioning RAID, LVM etc., disk related Management Tools Fdisk, Parted etc. , Boot Loaders GRUB, LILO, Custom Loaders, System administration – Role of system administrator, identifying administrative tasks & files,Configuration and log files, Chkconfig, Security Enhanced Linux, Installing and removing packages with rpm command

Reference Book :

1 . Official Red Hat Linux Users guide by Redhat, Wiley Dreamtech India
2. UNIX for programmers and users by Graham Glass & King Ables, Pearson Education
3. Beginning Linux Programming by Neil Mathew & Richard Stones, Wiley Dreamtech India
4. Red Hat Linux Bible by Cristopher Negus, Wiley Dreamtech India
5. UNIX Shell Programming by Yeswant Kanethkar, BPB
6. UNIX concepts and applications by sumitabha das, mcgraw hill publication

Learning Outcomes:

1. Describe the architecture and features of the Linux operating system and distinguish it from other operating systems.
2. Demonstrate Linux commands for file handling and process control.
3. Construct regular expressions for pattern matching and apply them to various filters for a specific task.
4. Analyze a given problem and apply requisite facets of shell programming in order to devise a shell script to solve the problem.

KBP-S-ECS-245 Probability Theory – II **Theory Lectures 30**

Learning Objectives :

1) Define event, outcome, trial, simple event, sample space and calculate the probability that an event will occur.
2) Calculate the probability of events for more complex outcomes.
3) Solve applications involving probabilities.

Unit - I Two dimensional discrete Random variable **(08 Lectures)**

Definition of two dimensional discrete r.v.,joint p.m.f.,marginal p.m.f., conditional p.m.f. ,Independence of two discrete r.v.
Expectation – definition, E(X+Y),E(XY) ,theorems on expectation , for two independent r.v.s,Cov(X,Y) Illustrative examples.

Unit – II Continuous Random Variable **(10 Lectures)**
Concept of continuous R.V. , probability density function(p.d.f.),cumulative distribution function(c.d.f/d.f.),properties of c.d.f. , Expectation of continuous r.v., Expectation of function continuous r.v.,variance,properties of mean and variance , Illustrative examples.

Unit – II **(12 Lectures)**

1) **Uniform distribution -** Definition ,notation,Real life situations mean and variance,c.d.f.,illustrative examples.
2) **Exponential distribution -** Definition ,notation, Real life situations mean and variance,c.d.f.,lack of memory property ,illustrative examples.
3) **Normal distribution -** Definition ,notation,mean and variance , Standard normal distribution,properties of normal distribution, approximation to binomial and Poisson distribution.
4) **Hypothesis Theorem**

Population, sample, random sample from distribution, parameter, statistic, standard error of estimator. Concept of null hypothesis, alternative hypothesis, critical Region, level of significance, type I & type II error, one sided & two sided tests.

Large sample test – Ho:μ = μo aginst H1, μ ≠ μo Ho:μ1 = μ2 aginst H1, μ1 ≠ μ2 Ho:P = Po aginst H1, P ≠ Po Ho:P1 = P2 aginst H1, P1 ≠ P2 Numerical problems.

Reference Books:

1. Fundamentals of mathematical statistics – Gupta and Kapoor
2. Modern elementary statistics – J.E. Fruend
3. Statistical Methods – j. Medhi
4. Programmed Statistics – B.L.Agarwal
5. Fundamental of Statistics – S.C.Gupta.

Learning outcomes

Students will be able to:

1. Calculate probabilities by applying probability laws and theoretical results.
2. Identify an appropriate probability distribution for a given discrete or continuous random variable and use its properties to calculate probabilities.
3. Derive probability distributions of functions of random variables.
4. Derive expressions for measures such as the mean and variance of common probability distributions using calculus and algebra.
5. Calculate probabilities for joint distributions including marginal and conditional probabilities.
6. For jointly distributed random variables calculate their covariance and correlation and determine whether they are independent.
7. Apply results from large-sample theory and the Central Limit Theorem to approximate a sampling distribution.
8. Implement basic simulation methods using statistical software to investigate sampling distributions.

KBP-S-ECS-246 Optimization Techniques **Theory Lectures 30**

Learning Objectives :

1. To understand the theory of optimization methods and algorithms developed for solving various types of optimization problems
2. To develop and promote research interest in applying optimization techniques in problems of Engineering and Technology
3. To apply the mathematical results and numerical techniques of optimization theory to concrete Engineering problems.

Unit - I **(10Lectures)**

Introduction to Operations Research: History, Evolution, Scope and Limitations.

Linear Programming Problem(LPP): Statement of LPP, formulation of problems as LPP, Definitions of Slack variables, surplus variables and artificial variable, standard form of LPP, Definitions of a solution, feasible solution, basic feasible solution and an optimum solution. Solution of LPP by graphical method and simplex method

Unit - II **(10Lectures)**

Transportation Problem (TP) : Statement of TP, balanced and unbalanced TP, methods of obtaining initial basic feasible solution of TP- North-West Corner method, method of matrix minima and Vogel's approximation method. MODI Method of obtaining an optimal solution of TP.

Unit - III **(10Lectures)**

Assignment Problem (AP): Statement of AP, balanced and unbalanced AP, relation with TP, Optimal solution of AP.A.P. for maximization for profit.

Job Sequencing Problem: Statement of problem, procedure of processing 'n' jobs on two and three machines, computation of elapsed time and idle time

Reference Books:

1) Operations Research - H.A.Taha
2) Operations Research - Kantiswarup Gupta
3) Linear Programming - S. Vajda

Learning Outcomes :

students should be able to

1.Cast engineering minima/maxima problems into optimization framework.

2.Learn efficient computational procedures to solve optimization problems.

3.Use Matlab to implement important optimization methods.

KBP-S-ECS-247 Programming Using Python **Theory Lectures 30**

Learning objectives :

1. To acquire programming skills in core Python.
2. To acquire Object Oriented Skills in Python
3. To develop the skill of designing Graphical user Interfaces in Python
4. To develop the ability to write database applications in Python

Unit - I **(08Lectures)**

Python Exception Handling and Regular Expression, File Input-Output:

Exception Handling: What is exception, Various keywords tohandle exception such try, catch, except, else, finally, raise

Regular Expressions: Concept of regular expression,various types of regular expressions, using match function

File Input-Output: Opening and closing file,Various types of file modes, Reading and writing to files,Manipulating directories

Unit -II **(06Lectures)**

GUI Programming in Python (using Tkinter/wxPython/Qt) : What is GUI, Advantages of GUI,Introduction to GUI library,Layout management ,Events and bindings, Font, Colors, drawing on Canvas (line, oval, rectangle, etc.).

Widget such as : Frame,Label,Button,Checkbutton,Entry, Listbox,Message, Radiobutton , Tex, Spinbox etc.

Unit -III **(06Lectures)**

Database connectivity in Python: Installing mysql server connector, Accessing connector module using connect, cursor,Execute& close functions,Reading single & multiple results of query execution, Executing different types of statements, Executing transactions, Understanding exceptions in database connectivity.

Unit -IV **(10Lectures)**

Flask Framework : Introduction, Installation, Application ,Routing , Variable rules,URL building ,HTTP methods, Templates ,static files, Request Object, Sending Form Data to template, Cookies, Session, Redirect & Errors ,Message Flashing, File Uploading, Extensions, Mail, WTF, SQLite, SQLAlchemy, Sijax, Deployment, FastCGI.

XML Introduction to XML: XML Parser Architecture and API’s, Parsing XML with SAX API’s, Parsing XML with DOM API’s

Reference Books:

1. Introduction to Computer Science using Python- Charles Dierbach
2. Beginning Python: Using Python 2.6 and Python
3- James Payne 3. Practical Programming: An Introduction toComputer Science Using Python 3- Paul Gries , Jennifer Campbell, Jason Montojo
4. Programming Languages – Principles and Paradigms- AdeshPandey
5. MySQL for Python: Database Access Made Easy- A. Lukaszewski

Learning Outcomes :

Student will be able to

1. Explain basic principles of Python programming language

2. Implement object oriented concepts,

3. Implement database and GUI applications.

KBP-S-ECS-248 Environmental studies **Theory Lectures 30**

Unit - I **(06Lectures)**

Introduction to environmental studies: Multidisciplinary nature of environmental studies; Scope and importance; Concept of sustainability and sustainable development.

Ecosystems:What is an ecosystem? Structure and function of ecosystem; Energy flow in an ecosystem: food chains, food webs and ecological succession. Case studies of the following ecosystems : a) Forest ecosystem b) Grassland ecosystem c) Desert ecosystem d) Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries)

Unit -II **(08Lectures)**

Natural Resources : Renewable and Non---renewable Resources ,Land resources and landuse change; Land degradation, soil erosion and desertification. Deforestation: Causes and impacts due to mining, dam building on environment, forests, biodiversity and tribal populations. Water: Use and over-exploitation of surface and ground water, floods, droughts, conflicts over water (international & inter---state). Energy resources : Renewable and non renewable energy sources, use of alternate energy sources, growing energy needs, case studies.

Biodiversity and Conservation: Levels of biological diversity : genetic, species and ecosystem diversity; Biogeographic zones of India; Biodiversity patterns and global biodiversity hot spots • India as a mega---biodiversity nation; Endangered and endemic species of India • Threats to biodiversity : Habitat loss, poaching of wildlife, man---wildlife conflicts, biological invasions; Conservation of biodiversity : In---situ and Ex---situ conservation of biodiversity. • Ecosystem and biodiversity services: Ecological, economic, social, ethical, aesthetic and Informational value.

Unit III **(08Lectures)**

EnvironmentalPollution: Environmental pollution : types, causes, effects and controls; Air, water, soil and noise pollution • Nuclear hazards and human health risks.Solid waste management : Control measures of urban and industrial waste. • Pollution case studies.

Environmental Policies & Practices: Climate change, global warming, ozone layer depletion, acid rain and impacts on human communities and agriculture

Environment Laws: Environment Protection Act; Air (Prevention & Control of Pollution) Act; Water (Prevention and control of Pollution) Act; Wildlife Protection Act; Forest Conservation Act. International agreements: Montreal and Kyoto protocols and Convention on Biological Diversity (CBD).Nature reserves, tribal populations and rights, and human wildlife conflicts in Indian context.

Unit-IV **(08Lectures)**

Human Communities and the Environment: Human population growth: Impacts on environment, human health and welfare.Resettlement and rehabilitation of project affected persons; case studies.

Disaster management : floods, earthquake, cyclones and landslides.
Environmental movements : Chipko, Silent valley, Bishnois of Rajasthan. •
Environmental ethics: Role of Indian and other religions and cultures in environmental conservation.Environmental communication and public awareness, case studies (e.g., CNG vehicles in Delhi).

Field work (Equal to 5 lectures) Visit to an area to document environmental assets: river/ forest/ flora/fauna, etc. Visit to a local polluted site---Urban/Rural/Industrial/Agricultural. Study of common plants, insects, birds and basic principles of identification. Study of simple ecosystems---pond, river, Delhi Ridge, etc.

KBP-S-ECS-LAB - I

Data Structure Practical

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. Use Template functions.
2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation. Use Templates.
8. Perform Queues operations using Circular Array implementation. Use Templates.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomial.
11. WAP to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
12. (ii) WAP to display fibonacci series (i)using recursion, (ii) using iteration
13. WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion
14. WAP to create a Binary Search Tree and include following operations in tree: (a) Insertion (Recursive and Iterative Implementation) (b) Deletion by copying (c) Deletion by Merging (d) Search a no. in BST (e) Display its preorder, postorder and inorder traversals Recursively (f) Display its preorder, postorder and inorder traversals Iteratively (g) Display its level-by-level traversals (h) Count the non-leaf nodes and leaf nodes (i) Display height of tree (j) Create a mirror image of tree (k) Check whether two BSTs are equal or not
15. WAP to convert the Sparse Matrix into non-zero form and vice-versa.
16. WAP to reverse the order of the elements in the stack using additional stack.
17. WAP to reverse the order of the elements in the stack using additional Queue.
18. WAP to implement Diagonal Matrix using one-dimensional array.
19. WAP to implement Lower Triangular Matrix using one-dimensional array.
20. WAP to implement Upper Triangular Matrix using one-dimensional array.

DBMS & Mysql Server Practical

1) SQL* formatting commands
2) To create a table, alter and drop table.
3) To perform select, update, insert and delete operation in a table.
4) To make use of different clauses viz where, group by, having, order by, union and intersection,
5) To study different constraints.

[SQL FUNCTION]

6) To use oracle function viz aggregate, numeric, conversion, string function.
7) To understand use and working with joins.
8) To make use of transaction control statement viz rollback, commit and save point.
9) To make views of a table.
10) To make indexes of a table.

[PL/SQL]

11) To understand working with PL/SQL
12) To implement Cursor on a table.
13) To implement trigger on a table

Python

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users choice.

2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria : Grade A: Percentage >=80 Grade B: Percentage>=70 and <80 Grade C: Percentage>=60 and <70 Grade D: Percentage>=40 and <60 Grade E: Percentage<40

3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input paramters from user.

4. WAP to display the first n terms of Fibonacci series.

5. WAP to find factorial of the given number.

6. WAP to find sum of the following series for n terms: 1 – 2/2! + 3/3! - - - - - n/n!

7. WAP to calculate the sum and product of two compatible matrices.

KBP-S-ECS-LAB - II

Software Testing

1. Write a program that take three inputs (a,b &c) that represent the sides of a triangle, and the output is one of the below four:
   a. Not a triangle b. Scalene triangle c. Isosceles triangle d. Equilateral triangle
   1.1 Generate test cases using Boundary Value Analysis, Equivalence Class Partitioning and Decision Table Testing.
   1.2 Generate test cases using Basis path testing.
   1.3 Run code coverage tool.
2. Write a program that determines the nature of roots of a quadratic equation. Output should be one of the following:-
   - Not a quadratic equation.
   - Complex roots
   - Real roots
   - Single roots

   I. Generate test cases using Boundary Value Analysis, Equivalence Class Partitioning and Decision Table Testing.
   II. Generate test cases using Basis path testing.
   III. Run code coverage tool
3. Write a program that checks whether the number is even or odd. Run code coverage tool and find the amount of code being covered.
4. Write a program that dynamically allocates memory to10 integers using malloc() or calloc() and
   - donot free memory leading to memory leaks. Verify the same usingValgrind.
   - Now, free memory using free() at the end of the program to avoid memory leaks. Verify the same using Valgrind.
5. Use LoadUI load testing tool to test the web application performance.

Software Engineering

Sr. No **Practical Title**

1. • Problem Statement, • Process Model
2. Requirement Analysis: • Creating a Data Flow • Data Dictionary, Use Cases
3. Project Management: • Computing FP • Effort • Schedule, Risk Table, Timeline chart
4. Design Engineering: • Architectural Design • Data Design, Component Level Design
5. Testing: • Basis Path Testing

Operating System & Linux & Shell Programming

1. Write a program (using *fork()* and/or *exec()* commands) where parent and child execute: a) same program, same code. b) same program, different code. - c) before terminating, the parent waits for the child to finish its task.
2. Write a program to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information)

3. Write a program to report behaviour of Linux kernel including information on configured memory, amount of free and used memory. (memory information)
4. Write a program to print file details including owner access permissions, file access time, where file name is given as argument.
5. Write a program to copy files using system calls.
6. Write program to implement FCFS scheduling algorithm.
7. Write program to implement Round Robin scheduling algorithm.
8. Write program to implement SJF scheduling algorithm.
9. Write program to implement non-preemptive priority based scheduling algorithm.
10. Write program to implement preemptive priority based scheduling algorithm.
11. Write program to implement SRJF scheduling algorithm.
12. Write program to calculate sum of n numbers using *thread* library.
13. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

Linux & Shell Programming Practical

1. Write a shell script to check if the number entered at the command line is prime or not.
2. Write a shell script to modify ―call‖ command to display calendars of the specified months.
3. Write a shell script to modify ―call‖ command to display calendars of the specified range of months.
4. Write a shell script to accept a login name. If not a valid login name display message – ―Entered login name is invalid‖.
5. Write a shell script to display date in the mm/dd/yy format.
6. Write a shell script to display on the screen sorted output of ―who‖ command along with the total number of users .
7. Write a shell script to display the multiplication table any number,
8. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
9. Write a shell script to find the sum of digits of a given number. 66
10. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
11. Write a shell script to find the LCD(least common divisor) of two numbers.
12. Write a shell script to perform the tasks of basic calculator.
13. Write a shell script to find the power of a given number.
14. Write a shell script to find the binomial coefficient C(n , x).
15. Write a shell script to find the permutation P(n,x).
16. Write a shell script to find the greatest number among the three numbers.
17. Write a shell script to find the factorial of a given number.

18. Write a shell script to check whether the number is Armstrong or not.
19. Write a shell script to check whether the file have all the permissions or not.

Python:Visual Python

All the programs should be written using user defined functions, wherever possible.

1. Write a menu-driven program to create mathematical 3D objects I. curve II. sphere III. cone IV. arrow V. ring VI. cylinder.
2. WAP to read n integers and display them as a histogram.
3. WAP to display sine, cosine, polynomial and exponential curves.
4. WAP to plot a graph of people with pulse rate p vs. height h. The values of p and h are to be entered by the user.
5. WAP to calculate the mass m in a chemical reaction. The mass m (in gms) disintegrates according to the formula m=60/(t+2), where t is the time in hours. Sketch a graph for t vs. m, where t>=0.
6. A population of 1000 bacteria is introduced into a nutrient medium. The population p grows as follows: P(t) = (15000(1+t))/(15+ *e)* where the time t is measured in hours. WAP to determine the size of the population at given time t and plot a graph for P vs t for the specified time interval.
7. Input initial velocity and acceleration, and plot the following graphs depicting equations of motion: I. velocity wrt time (v=u+at) II. distance wrt time (s=u*t+0.5*a*t*t) III. distance wrt velocity (s=(v*v-u*u)/2*a)
8. WAP to show a ball bouncing between 2 walls.

KBP-S-ECS-LAB - III

STATISTICS PRACTICAL

Important note: - All the practical's will be conducted using Python & R-Programming.

1) Fitting of Poisson disuibution
2) Fitting of Binomial distribution
3) Fining of)vet zeommic distribution
4) Model sampling from Poisson distribution
5) Model sampling from Binomial distribution
6) Model sampling from Poisson distribution
7) Two dimensional discrete r v
8) Fitting of continuous Uniform distribution
9) Fining of Exponential distribution.
10) Fitting of Normal distribution
11) Model sampling from uniform and exponential distribution.
12) Model sampling from Normal distribution
13) Large sample tests
14) Graphical and diagrammatic representation
15) Measures of central tendency (Grouped and ungrouped data)
16) Measures of Dispersion
17) Correlation .
18) Regression
19) Multiple regression.
20)Multiple and partial correlation.

Optiniizaltion Techniques List of experiments:

Sr.No Name of Experiment

1) Solution of LPP using Graphical method.
2) Solution of LPP us, Simplex method.
3) Transportation problem — I (Initial Basic Feasible Solution)
4) Ttansportation problem — (Optimum solution)
5) Assignment Problem
6) Sequencing problem.