



# **Savitribai Phule Pune University**

(Formerly University of Pune)

Three Year B.Sc. Degree Program in Computer Science

(Faculty of Science & Technology)

## **T.Y.B.Sc. (Computer Science)**

Choice Based Credit System Syllabus

To be implemented from Academic Year 2021 – 2022

**Course Structure T.Y.B.Sc. (Computer Science)**

**Semester V (Total credits=22)**

Course type	Paper Code	Paper title	Credits		Evaluation		
			T	P	CA	UA	TOTAL
DSEC - I	CS-351	Operating Systems – I	2		15	35	50
	CS-352	Computer Networks – II	2		15	35	50
	CS-357	Practical course based on CS 351		2	15	35	50
DSEC - II	CS-353	Web Technologies – I	2		15	35	50
	CS-354	Foundations of Data Science	2		15	35	50
	CS-358	Practical course based on CS 353 and CS 354		2	15	35	50
DSEC - III	CS-355	Object Oriented Programming using Java - I	2		15	35	50
	CS-356	Theoretical Computer Science	2		15	35	50
	CS-359	Practical Course based on CS 355		2	15	35	50
SECC - I	CS-3510	Python Programming	2	0	15	35	50
SECC - II	CS-3511	Blockchain Technology	2	0	15	35	50

**Semester VI (Total credits=22)**

Course type	Paper Code	Paper title	Credits		Evaluation		
			T	P	CA	UA	TOTAL
DSEC - I	CS-361	Operating Systems – II	2		15	35	50
	CS-362	Software Testing	2		15	35	50
	CS-367	Practical course based on CS 361		2	15	35	50
DSEC - II	CS-363	Web Technologies – II	2		15	35	50
	CS-364	Data Analytics	2		15	35	50
	CS-368	Practical course based on CS 363 and CS 364		2	15	35	50
DSEC - III	CS-365	Object Oriented Programming using Java - II	2		15	35	50
	CS-366	Compiler Construction	2		15	35	50
	CS-369	Practical Course based on CS 365		2	15	35	50
SECC - III	CS-3610	Software Testing Tools	2	0	15	35	50
SECC - IV	CS-3611	Project	2	0	15	35	50

**SavitribaiPhule Pune University**  
**T.Y.B.Sc. (Computer Science) - Sem – V**  
**Course Type: DSEC – I                      Course Code : CS - 351**  
**Course Title : Operating Systems – I**

Teaching Scheme: 03 Lect / week	No. of Credits: 2	Examination Scheme: IE : 15 marks UE: 35 marks
------------------------------------	----------------------	--

**Prerequisites**  
Data structures like stack, queue, linked list, tree, graph, hashing, file structures, any structured programming language

**Course Objectives:**  
1. To understand the concept of operation system and its principle  
2. To study the various functions and services provided by operating system  
3. To understand the notion of process and threads

**Course Outcomes: After completion of this course students will be able to understand the concept of**  
1. Processes and Thread Scheduling by operating system  
2. Synchronization in process and threads by operating system  
3. Memory management by operating system using with the help of various schemes

**Course Contents**

<b>Chapter 1</b>	<b>Introduction to Operating Systems</b>	<b>6 lectures</b>
	<ul style="list-style-type: none"> <li>• Operating Systems Overview- system Overview and Functions of operating systems</li> <li>• What does an OS do?</li> <li>• Operating system Operations</li> <li>• Operating system structure</li> <li>• Protection and security</li> <li>• Computing Environments- Traditional, mobile , distributed, Client/server, peer to peer computing</li> <li>• Open source operating System</li> <li>• Booting</li> <li>• Operating System services,</li> <li>• System calls Types of System calls and their working.</li> </ul>	
<b>Chapter 2</b>	<b>Processes and Threads</b>	<b>6 lectures</b>
	<ul style="list-style-type: none"> <li>• Process Concept – The processes, Process states, Process control block.</li> <li>• Process Scheduling – Scheduling queues, Schedulers, context switch</li> <li>• Operations on Process – Process creation with program using fork(), Process termination</li> <li>• Thread Scheduling- Threads, benefits, Multithreading Models, Thread Libraries</li> </ul>	
<b>Chapter 3</b>	<b>Process Scheduling</b>	<b>7 lectures</b>
	<ul style="list-style-type: none"> <li>• Basic Concept – CPU-I/O burst cycle, Scheduling Criteria ,CPU scheduler, Preemptive scheduling, Dispatcher</li> <li>• Scheduling Algorithms – FCFS, SJF, Priority scheduling, Round-robin scheduling, Multiple queue scheduling, Multilevel feedback queue scheduling</li> </ul>	
<b>Chapter 4</b>	<b>Synchronization</b>	<b>5 lectures</b>
	<ul style="list-style-type: none"> <li>• Background</li> <li>• Critical Section Problem</li> <li>• Semaphores: Usage, Implementation</li> </ul>	

<ul style="list-style-type: none"> <li>• Classic Problems of Synchronization – The bounded buffer problem, The reader writer problem, The dining philosopher problem</li> </ul>		
<b>Chapter 5</b>	<b>Memory Management</b>	<b>12 lectures</b>
<ul style="list-style-type: none"> <li>• Background – Basic hardware, Address binding, Logical versus physical address space, Dynamic loading, Dynamic linking and shared libraries</li> <li>• Swapping</li> <li>• Contiguous Memory Allocation – Memory mapping and protection, Memory allocation, Fragmentation</li> <li>• Paging – Basic Method, Hardware support, Protection, Shared Pages</li> <li>• Segmentation – Basic concept, Hardware</li> <li>• Virtual Memory Management – Background, Demand paging, Performance of demand paging, Page replacement – FIFO, Optimal, LRU, MFU</li> </ul>		
<b>Reference Books:</b>		
<ol style="list-style-type: none"> <li>1. Operating System Concepts, Avi Silberschatz, Peter Galvin, Greg Gagne, Student Edition, Wiley Asia</li> <li>2. Operating Systems: Internals and Design Principles, William Stallings, Prentice Hall of India.</li> <li>3. Advanced Concepts in Operating Systems, M Singhal and NG Shivaratri, Tata McGraw Hill Inc, 2001</li> <li>4. The ‘C’ Odyssey, UNIX-the open boundless C, Meeta Gandhi, Tilak Shetty, Rajiv Shah, BPB publication</li> </ol>		

<p style="text-align: center;"><b>SavitribaiPhule Pune University</b>  <b>T.Y.B.Sc. (Computer Science) Sem - V</b>  <b>Course Code: DSEC - I                      Course Code : CS - 352</b>  <b>Course Title :Computer Networks - II</b></p>		
Teaching Scheme 03 Lect/ week	No. of Credits 2	Examination Scheme IE :15 marks UE: 35 marks
<p><b>Prerequisites:</b> Prerequisites: Basic knowledge of Networking and ISO/OSI model</p>		
<p><b>Course Objectives</b></p> <ul style="list-style-type: none"> <li>• To understand different protocols of application layer.</li> <li>• To understand concepts of multimedia.</li> <li>• Explore the different methods used for Network/INTERNET security.</li> </ul>		
<p><b>Course Outcomes</b> On completion of the course, student will be able to–</p> <ul style="list-style-type: none"> <li>• Student will understand the different protocols of Application layer.</li> <li>• Develop understanding of technical aspect of Multimedia Systems</li> <li>• Develop various Multimedia Systems applicable in real time.</li> <li>• Identify information security goals.</li> <li>• Understand, compare and apply cryptographic techniques for data security.</li> </ul>		
<b>Course Contents</b>		
<b>Chapter 1</b>	<b>Application Layer</b>	<b>10 Lect</b>
<p>Domain Name System</p> <ul style="list-style-type: none"> <li>• Name space-Flat name space, Hierarchical name space</li> <li>• Domain Name Space -Label ,Domain name, FQDN,PQDN</li> <li>• Distribution of Domain Name Space-Hierarchy of name servers, zone, Root server, Primary and secondary servers.</li> <li>• DNS in the Internet: Generic domains, Country domains,inverse domain</li> <li>• Resolution-Resolver,mapping names to address,mapping addresses to names,recursive resolution,iterative resolution,caching</li> </ul> <p>Electronic Mail-</p> <ul style="list-style-type: none"> <li>• Architecture-First scenario, second scenario, Third scenario, Fourth scenario</li> <li>• User agent-services of user agent, types of UA Format of e-mail</li> <li>• MIME-MIME header</li> <li>• Message transfer agent-SMTP</li> <li>• Message Access Agent: POP and IMAP</li> </ul> <p>File Transfer</p> <p>FTP-Communication over data control connection,File type,data structure,Transmission mode,anonymous FTP</p>		
<b>Chapter 2</b>	<b>Multimedia</b>	<b>08 Lect</b>
<p>Digitizing audio and video, Audio and Video compression</p> <p>Streaming Stored audio/video</p> <ul style="list-style-type: none"> <li>• First approach</li> <li>• Second approach</li> <li>• Third approach</li> <li>• Fourth approach</li> </ul> <p>Streaming live audio/video</p>		

<p>Real time interactive audio/video- Characteristics, Time relationship, timestamp, Playback buffer, ordering multicasting, translation</p> <p>RTP-Packet format</p> <p>RTCP-Message types</p> <p>Voice over IP-SIP,SIP sessionH.323-Architecture, Protocols</p>	
<b>Chapter 3</b>	<p><b>Cryptography and Network Security</b></p> <p><b>09 Lect</b></p>
<p>Terminology: Cryptography, plain text and cipher text, cipher key, categories of cryptography-Symmetric key, asymmetric key</p> <p>Encryption model</p> <p>Symmetric key cryptography</p> <ul style="list-style-type: none"> <li>• Traditional ciphers – substitution cipher, shift cipher, Transposition cipher</li> <li>• Simple Modern ciphers-XOR, Rotation cipher, s-box,p-box</li> <li>• Modern round ciphers-DES</li> <li>• Mode of operation-ECB,CBC,CFB,OFB</li> </ul> <p>Asymmetric key cryptography-RSA</p> <p>Security Services</p> <ul style="list-style-type: none"> <li>• Message confidentiality-With Symmetric key cryptography, with asymmetric key cryptography</li> <li>• Message integrity-Document and fingerprint, message and message digest</li> <li>• Message authentication-MAC,HMAC</li> <li>• Digital signature</li> <li>• Entity Authentication-Passwords, Fixed passwords challenge-response</li> </ul>	
<b>Chapter 4</b>	<p><b>Security in the Internet</b></p> <p><b>09 Lect</b></p>
<p>IPSecurity(IPSec)</p> <ul style="list-style-type: none"> <li>• Two modes</li> <li>• Two security protocols</li> <li>• Services provided by IPSec</li> <li>• Security association</li> <li>• Internet key exchange</li> <li>• Virtual private network</li> </ul> <p>SSL/TLS</p> <ul style="list-style-type: none"> <li>• SSL services</li> <li>• Security parameters</li> <li>• Sessions and connections</li> <li>• Four protocols</li> <li>• Transport layer security</li> </ul> <p>PGP</p> <ul style="list-style-type: none"> <li>• Security parameters</li> <li>• Services</li> <li>• PGP algorithms</li> <li>• Key rings</li> <li>• PGP certificates</li> </ul> <p>Firewalls</p> <ul style="list-style-type: none"> <li>• Packet filter firewall</li> <li>• Proxy firewall</li> </ul>	

**Reference Books:**

1. Data communications and networking by Behrouz Forouzan 4<sup>th</sup>/5<sup>th</sup> edition, McGraw Hill Pvt Ltd.
2. Computer Networks by Andrew S Tanenbaum, 4<sup>th</sup>/5<sup>th</sup> edition, Pearson Education
3. Cryptography and Network Security: Principles and Practice, William Stallings, 7<sup>th</sup> edition, Pearson Education
4. Network Security Essentials: Applications and Standards (For VTU), William Stallings, 3<sup>rd</sup> edition, Pearson Education





Random access to file data  
Getting information on file  
Ownership and permissions  
Using PHP to access a database  
Relational databases and SQL  
PEAR DB basics  
Advanced database techniques

**Chapter 5 | Handling email with php**

**2 Lectures**

Email background  
Internet mail protocol  
Structure of an email message  
Sending email and validation of Email\_id with php

**Reference Books:**

1. HTML & CSS: The Complete Reference, Fifth Edition Author: Thomas A. Powell  
First published: 01 Jan 2010.
2. Programming PHP By Rasmus Lerdorf and Kevin Tatroe, O'Reilly publication
3. Beginning PHP 5 , Wrox publication
4. PHP web seVICES, Wrox publication
5. Mastering PHP , BPB Publication
6. PHP cookbook, O'Reilly publication
7. PHP for Beginners, SPD publication
8. Programming the World Wide Web , Robert W Sebesta(3rd Edition)
9. HTML 5 Black Book : Covers Css3, Javascript, XML, XHTML, Ajax, PHP And JQuery by  
Kogent Learning Solutions Inc, Published November 2011 by Dreamtech Press
10. Spurlock Jake, Bootstrap: Responsive Web development. O'Reilly Media, Inc

Ref. Links

11. [www.php.net.in](http://www.php.net.in)
12. [www.W3schools.com](http://www.W3schools.com)
13. [www.wrox.com](http://www.wrox.com)
14. <https://coreui.io/docs/layout/grid/#grid-options>
15. <https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php>



<b>Chapter 2</b>	<b>Statistical Data Analysis</b>	<b>10 lectures</b>
<p>2.1.Role of statistics in data science</p> <p>2.2.Descriptive statistics</p> <p style="padding-left: 20px;">Measuring the Frequency</p> <p style="padding-left: 20px;">Measuring the Central Tendency: Mean, Median, and Mode</p> <p style="padding-left: 20px;">Measuring the Dispersion: Range, Standard deviation, Variance, Interquartile Range</p> <p>2.3.Inferential statistics</p> <p style="padding-left: 20px;">Hypothesis testing, Multiple hypothesis testing, Parameter Estimation methods,</p> <p>2.4.Measuring Data Similarity and Dissimilarity</p> <p style="padding-left: 20px;">Data Matrix versus Dissimilarity Matrix, Proximity Measures for Nominal Attributes, Proximity Measures for Binary Attributes, Dissimilarity of Numeric Data: Euclidean, Manhattan, and Minkowski distances, Proximity Measures for Ordinal Attributes</p> <p>2.5.Concept of Outlier, types of outliers, outlier detection methods</p>		
<b>Chapter 3</b>	<b>Data Preprocessing</b>	<b>10 lectures</b>
<p>Data Objects and Attribute Types: What Is an Attribute?, Nominal , Binary, Ordinal Attributes, Numeric Attributes, Discrete versus Continuous Attributes</p> <p style="padding-left: 20px;">Data Quality: Why Preprocess the Data?</p> <p>3.3.Data munging/wrangling operations</p> <p>Cleaning Data - Missing Values, Noisy Data (Duplicate Entries, Multiple Entries for a Single Entity, Missing Entries, NULLs, Huge Outliers, Out-of-Date Data, Artificial Entries, Irregular Spacings, Formatting Issues - Irregular between Different Tables/Columns, Extra Whitespace, Irregular Capitalization, Inconsistent Delimiters, Irregular NULL Format, Invalid Characters, Incompatible Datetimes)</p> <p>Data Transformation – Rescaling, Normalizing, Binarizing, Standardizing,Label and One Hot Encoding</p> <p>Data reduction</p> <p>Data discretization</p>		
<b>Chapter 4</b>	<b>Data Visualization</b>	<b>10 lectures</b>
<p>Introduction to Exploratory Data Analysis</p> <p>Data visualization and visual encoding</p> <p>Data visualization libraries</p> <p>Basic data visualization tools</p> <p style="padding-left: 20px;">Histograms, Bar charts/graphs, Scatter plots, Line charts, Area plots, Pie charts, Donut charts</p> <p>Specialized data visualization tools</p> <p style="padding-left: 20px;">Boxplots, Bubble plots, Heat map, Dendrogram, Venn diagram, Treemap, 3D scatter plots</p> <p style="padding-left: 20px;">Advanced data visualization tools- Wordclouds</p> <p style="padding-left: 20px;">Visualization of geospatial data</p> <p style="padding-left: 20px;">Data Visualization types</p>		
<b>Reference Books:</b>		
<ol style="list-style-type: none"> <li>1) Data Science Fundamentals and Practical Approaches, Gypsy Nandi, Rupam Sharma, BPB Publications, 2020.</li> <li>2) The Data Science Handbook, Field Cady, John Wiley &amp; Sons, Inc, 2017</li> <li>3) Data Mining Concepts and Techniques, Third Edition, Jiawei Han, Micheline</li> </ol>		

Kamber, Jian Pei, Morgan Kaufmann, 2012.

4) A Hands-On Introduction to Data Science, Chirag Shah, University of  
Washington Cambridge University Press



<b>Chapter 3</b>	<b>Inheritance and Interface</b>	<b>8 Lect</b>
<p>Inheritance Basics (extends Keyword) and Types of Inheritance  Superclass, Subclass and use of Super Keyword  Method Overriding and runtime polymorphism  Use of final keyword related to method and class  Use of abstract class and abstract methods  Defining and Implementing Interfaces  Runtime polymorphism using interface  Concept of Marker and Functional Interfaces</p>		
<b>Chapter 4</b>	<b>Exception and File Handling</b>	<b>5 Lect</b>
<p>Dealing with errors , Exception class, Checked And Unchecked Exception  Catching Exceptions, Multiple Catch Block, Nested try block  Creating User Defined Exception  Introduction to Files And Streams  Input-OutputStream : FileInputStream/OutputStream, BufferedInput/OutputStream,  DataInput/OutputStream  Reader-Writer : FileReader/Writer, BufferedReader/Writer,  InputStreamReader, OutputStreamWriter</p>		
<b>Chapter 5</b>	<b>User Interface with AWT and Swing</b>	<b>10 Lect</b>
<p>What is AWT? What is Swing? Difference between AWT and Swing  The MVC Architecture And Swing  Layouts And Layout Managers  Containers And Components – JFrame, JButton, JLabel, JText, JTextArea,  JCheckBox And JRadioButton, JList, JComboBox, JMenu And related Classes  Dialogs (Message, Confirmation, Input), JFileChooser, JColorChooser  Event Handling: Event Sources, Listeners  Adapters And Anonymous Inner Class</p>		
<b>Reference Books:</b>		
<p>R1. Complete reference Java by Herbert Schildt(5th edition)  R2. Java 2 programming black books, Steven Horlzner  R3. Programming with Java , A primer ,Forth edition , By E. Balagurusamy  R4. Core Java Volume-I-Fundamentals, Eighth Edition, Cay S. Horstmann, Gary Cornell,  Prentice Hall, Sun Microsystems Pres</p>		

<p style="text-align: center;"><b>Savitribai Phule Pune University</b>  <b>T.Y.B.Sc. (Computer Science) Sem - V</b>  <b>Course Type: DSEC - III</b>                      <b>Course Code: CS - 356</b>  <b>Paper Title: Theoretical Computer Science</b></p>		
Teaching Scheme 3 Lect/ week	No. of Credits 2	Examination Scheme IE : 15 marks UE: 35 marks
<p><b>Prerequisites</b></p> <ul style="list-style-type: none"> <li>• Mathematical Preliminaries Sets (Subset, Set Operations), Relations (Properties of Relations, Closure of Relations) and Functions</li> <li>• Discrete Mathematics- Graphs, Trees, Logic and Proof Techniques</li> </ul>		
<p><b>Course Objectives</b></p> <ul style="list-style-type: none"> <li>• To understand the Finite Automata, Pushdown Automata and Turing Machine.</li> <li>• To understand the Regular Language, Context Free Language, Context Sensitive Language and Unrestricted Language.</li> <li>• To understand the relation between Automaton and Language</li> </ul>		
<p><b>Course Outcomes</b></p> <p>On completion of the course, student will be able to–</p> <ul style="list-style-type: none"> <li>• Understand the use of automata during language design.</li> <li>• Relate various automata and Languages.</li> </ul>		
<b>Course Contents</b>		
<b>Chapter 1</b>	<b>Finite Automaton</b>	<b>10 Lect</b>
<p>Introduction: Symbol, Alphabet, String, Prefix &amp; Suffix of Strings, Formal Language, Operations on Languages.  Deterministic finite Automaton – Definition, DFA as language recognizer, DFA as pattern recognizer.  Nondeterministic finite automaton – Definition and Examples.  NFA To DFA (Myhill Nerode Method)  NFA with <math>\epsilon</math>-transitions Definition and Examples.  NFA with <math>\epsilon</math>-Transitions to DFA &amp; Examples  Finite automaton with output – Mealy and Moore machine, Definition and Examples.  Minimization of DFA, Algorithm &amp; Problem using Table Method.</p>		
<b>Chapter 2</b>	<b>Regular Expressions and Languages</b>	<b>6 Lect</b>
<p>Regular Expressions (RE): Definition &amp; Example  Regular Expressions Identities.  Regular language-Definition and Examples.  Conversion of RE to FA-Examples.  Pumping lemma for regular languages and applications.  Closure Properties of regular Languages</p>		

<b>Chapter 3</b>	<b>Context-Free Grammars and Languages</b>	<b>10 Lect</b>
<p>Grammar - Definition and Examples.  Derivation-Reduction - Definition and Examples.  Chomsky Hierarchy.  CFG: Definition &amp; Examples. LMD, RMD, Parse Tree  Ambiguous Grammar: Concept &amp; Examples.  Simplification of CFG: Removing Useless Symbols, Unit Production, <math>\epsilon</math>-production and Nullable Symbol.  Normal Forms: Greibach Normal Form (GNF) and Chomsky Normal Form (CNF)  Regular Grammar: Definition.  Left linear and Right Linear Grammar-Definition and Example.  Equivalence of FA &amp; Regular Grammar  Construction of regular grammar equivalent to a given DFA.  Construction of a FA from the given right linear grammar</p>		
<b>Chapter 4</b>	<b>Push Down Automata</b>	<b>5 Lect</b>
<p>Definition of PDA and examples.  Construction of PDA using empty stack and final State method: Examples using stack method.  Definition DPDA &amp; NPDA, their correlation and Examples of NPDA  CFG (in GNF) to PDA: Method and examples</p>		
<b>Chapter 5</b>	<b>Turing Machine</b>	<b>5 Lect</b>
<p>The Turing Machine Model, Definition and Design of TM  Problems on language recognizers.  Language accepted by TM.  Types of Turing Machines (Multitrack TM, Two-way TM, Multitape TM, Non-deterministic TM)  Introduction to LBA (Basic Model) &amp; CSG. (Without Problems)</p>		
<b>Reference Books</b>		
<ol style="list-style-type: none"> <li>1. Introduction to Automata Theory, Languages and Computation, John E. Hopcraft, Rajeev Motwani, Jeffrey D. Ullman, Third Edition, Pearson Education Publication, 2008</li> <li>2. Introduction to Automata theory, Languages and computation By John E. Hopcroft and Jeffrey Ullman – Narosa Publishing House, 1995</li> <li>3. Theory of Computer Science Automata, Languages and Computation, K.L.P. Mishra, N. Chandrasekaran, Publication- Prentice Hall of India, 2008</li> <li>4. Introduction to Computer Theory Daniel I. A. Cohen – 2<sup>nd</sup> edition – John Wiley &amp; Sons, 1996</li> <li>5. Introduction to Languages and The Theory of Computation John C. Martin The McGraw-Hill, Fourth Edition, 2011</li> </ol>		



**Savitribai Phule Pune University**  
**T.Y.B.Sc. (Computer Science) - Sem - V**  
**Course Type: DSEC - I                      Course Code: CS - 357**  
**Course Title : Practical Course based on CS - 351**

Teaching Scheme:  
5 Lect/ week

No. of Credits:  
2

Examination Scheme:  
IE : 15 marks  
UE: 35 marks

**Course Objectives:**

1. To understand the concept of process scheduling with the help of simulation.
2. To study the concept demand paging concepts in operating system.
3. To understand the working of operating system shell.

**Course Outcomes: After completion of this course students will be able to understand the concept of**

1. Process synchronization
2. Processes and Thread Scheduling by operating system
3. Memory management by operating system using with the help of various schemes

**Guidelines:**

1. Operating system platform – Linux
2. Programming language - C

**List of Assignments:**

- Operations on processes : (2 slot)  
(Create a child process using fork() and commands like exec(),execv() and execvp())
- Simulation of Operating System Shell and its working (commands)(2 slots)
- Simulation of CPU Scheduling Algorithms – FCFS, SJF, Priority and Round Robin(4 slots)
- Simulation of demand paging using memory page replacement algorithms – FIFO, LRU, OPT, MFU(4 slots)



**Assignment 4: Data preprocessing**

Apply data preprocessing techniques that are likely required for the dataset.

1) Partition them into appropriate number of bins by equal-frequency as well as equal-width partitioning.

2) Use smoothing by bin means to smooth the data based on the above partitioning,

3) Normalize the attribute based on min-max normalization and z-score normalization.

Comment on which method you would prefer to use for partitioning, smoothing, and normalization for the given attribute.

**Assignment 5: Data Visualization with matplotlib**

View the data using various 2-D, 3-D plots and charts, setting styles, saving the figures, customizing the legends, multiple subplots,

**SavitribaiPhule Pune University**  
**T.Y.B.Sc. (Computer Science) - Sem – V**  
**Course Type: DSEC - III**                      **Course Code: CS - 359**  
**Course Title : Practical Course based on CS - 355**

Teaching Scheme 5 Lect / week Batch Size : 12	No. of Credits 2	Examination Scheme IE : 15 marks UE: 35 marks
---	---------------------	---

**Course Objectives:**

Covers the complete scope of the syllabus.

1. Bringing uniformity in the way course is conducted across different colleges.
2. Continuous assessment of the students.

**Course Outcomes:**

1. Use an integrated development environment to write, compile, run, and test simple object-oriented Java programs.
2. Read and make elementary modifications to Java programs that solve real-world problems.
3. Validate input in a Java program.

**Guidelines:**

**Operating Environment :**

- **Operating system :** Linux
- **Editor :** Anylinux based editor like vi, gedit and Use of IDE – Eclipse etc.
- **Compiler :** javac

**Submission :**

Each assignment will be assessed on a scale of 0 to 5 as indicated below.

- Not done                      0
- Incomplete                    1
- Late Complete                2
- Needs improvement        3
- Complete                      4
- Well Done                     5

**Assessment :**

Easy : All exercises are compulsory.

Medium : All exercises are compulsory.

**List of Assignments :**

**Assignment 1 : Java Tools and IDE, Simple java programs [Slot – 1]**

Introduction to the java environment  
Use of java tools like java, javac, jdb and javadoc  
Defining simple classes and creating objects.

**Assignment 2 : Array of Objects and Packages [Slot – 2]**

Defining a class.  
Creating an array of objects.  
Creating a package.

**Assignment 3 : Inheritance and Interfaces [Slot – 2]**

To implement inheritance in java.

To define abstract classes.

To define and use interfaces and Functional Interface.

**Assignment 4 : Exception And File Handling [Slot – 2]**

Demonstrate Exception Handling Mechanism in Java.

Use of try, catch, throw, throws ,finally blocks

Defining User defined Exception classes.

Creation of files and demonstration of I-O operations

**Assignment 5 : GUI Designing, Event Handling [Slot – 5]**

To demonstrate GUI creation using Swing Package and Layout managers.

To understand Event handling mechanism in Java.

Using Event classes, Event Listeners and Adapters

**Savitribai Phule Pune University**  
**T.Y.B.Sc. (Computer Science) - Sem – V**  
**Course Type: SECC – I                      Course Code : CS-3510**  
**Course Title: Python Programming**

Teaching Scheme 03 Lect / week	No. of Credits 2	Examination Scheme: IE : 15 marks UE: 35 marks
-----------------------------------	---------------------	--

**Course Objectives**

1. To introduce programming concepts using python
2. Student should be able to develop Programming logic using python
3. To develop basic concepts and terminology of python programming
4. To test and execute python programs

**Course Outcomes**

On completion of the course, student will be able to–

- Develop logic for problem solving
- Determine the methods to create and develop **Python programs** by utilizing the data structures like lists, dictionaries, tuples and sets.
- To be familiar about the basic constructs of programming such as data, operations, conditions, loops, functions etc.
- To write python programs and develop a small application project

**Course Contents**

<b>Chapter 1</b>	<b>An Introduction to Python</b>	<b>3 Lect</b>
------------------	----------------------------------	---------------

Introduction to Python

The Python Programming Language, History, features, Applications, Installing Python, Running Simple Python program

Basics of Python

Standard data types - basic, none, Boolean (true & False), numbers, Variables, Constants, Python identifiers and reserved words, Lines and indentation, multi-line statements and Comments, Input/output with print and input , functions Declaration, Operations on Data such as assignment, arithmetic, relational, logical and bitwise operations, dry run, Simple Input and output etc.

<b>Chapter 2</b>	<b>Control Statements</b>	<b>4 Lect</b>
------------------	---------------------------	---------------

**Sequence Control** – Precedence of operators, Type conversion

**Conditional Statements:** if, if-else, nested if-else,

**Looping-** for, while, nested loops, loop control statements (break, continue, pass)

a. **Strings:** declaration, manipulation, special operations, escape

character, string formatting operator, Raw String, Unicode strings, Built-in String methods.

<b>Chapter 3</b>	<b>Lists, functions, tuples and dictionaries, Sets</b>	<b>7 Lect</b>
------------------	--	---------------

**Python Lists:** Concept, creating and accessing elements, updating & deleting lists, traversing a List, reverse Built-in List Operators, Concatenation, Repetition, In Operator, Built-in List functions and methods.

**Functions:** Definitions and Uses, Function Calls, Type Conversion Functions, Math Functions, Composition, Adding New Functions, Flow of Execution, Parameters and Arguments, Variables and Parameters, Stack Diagrams, Void Functions, Anonymous functions Importing with from, Return Values, Boolean Functions, More Recursion, Functional programming tools - filter(), map(), and reduce(), recursion, lambda forms.

**Tuples and Dictionaries:** Tuples, Accessing values in Tuples, Tuple Assignment, Tuples as return values, Variable-length argument tuples, and Basic tuples operations, Concatenation, Repetition, in Operator, Iteration, Built-in tuple functions, indexing, slicing and matrices. Creating a Dictionary, Accessing Values in a dictionary, Updating Dictionary, Deleting Elements from Dictionary, Properties of Dictionary keys, Operations in Dictionary, Built-In Dictionary Functions, Built-in Dictionary Methods.

**Sets-** Definition, transaction of set(Adding, Union, intersection), working with sets

<b>Chapter 4</b>	<b>Modules ,Working with files, Exception handling</b>	<b>4 Lect</b>
------------------	--	---------------

**Modules:** Importing module, Creating & exploring modules, Math module, Random module, Time module

**Packages:** Importing package, creating package,examples

**Working with files:** Creating files and Operations on files (open, close, read, write), File object attributes, file positions, Listing Files in a Directory, Testing File Types, Removing files and directories, copying and renaming files, splitting pathnames, creating and moving directories

**Regular Expression-** Concept of regular expression, various types of regular expressions, using match function.

**Exception Handling:** Built-in Exceptions, Handling Exceptions, Exception with Arguments, User-defined Exceptions.

<b>Demonstration</b>	<b>Programming Assignments:</b>	<b>18 Lect</b>
----------------------	---------------------------------	----------------

**Out of 36 lectures, 18 are assigned for demonstration. Teacher should give demonstration of various programs mentioned below in the classroom or in the laboratory as per their convenience.**

Programming assignments should be done individually by the student in their respective login from the list given in Labbook. The codes should be uploaded on either the local server, Moodle, Github or any LMS.

Assignment 1 - Python Basics

Assignment 2 – Arrays, Strings, and Functions

Assignment 3 - List, Tuples, Sets, and Dictionary

Assignment 4 - File Handling and Date-Time

Assignment 5 - Exception handling and Regular expression

**Reference Books:**

1. An Introduction to Computer Science using Python 3 by Jason Montojo, Jennifer Campbell, Paul Gries, The pragmatic bookshelf-2013
2. James Payne, “Beginning Python: Using Python and Python 3.1,Wrox Publication
3. Introduction to Computer Science Using Python- Charles Dierbach, Wiley Publication Learning with Python “, Green Tea Press, 2002
4. Introduction to Problem Solving with Python by E balguruswamy,TMH publication- 2016
5. Beginning Programming with Python for Dummies Paperback – 2015 by John Paul Mueller
6. Object-oriented Programming in Python, Michael H. Goldwasser, David Letscher, Pearson Prentice Hall-2008

**Savitribai Phule Pune University**  
**T.Y.B.Sc. (Computer Science) - Sem – V**  
**Course Type: SECC – II                      Course Code : CS-3511**  
**Course Title: Blockchain Technology**

Teaching Scheme 03 Lect / week	No. of Credits 2	Examination Scheme: IE : 15 marks UE: 35 marks
-----------------------------------	---------------------	--

**Prerequisite:**  
Understanding of Object Oriented Programming Concepts  
Knowledge of Python

- Course Objectives**
1. Understand what and why of blockchain technology.
  2. Explore major components of blockchain.
  3. Learn about Bitcoin, Cryptocurrency and Ethereum.
  4. To learn blockchain programming using Python, Flask Web Framework, and HTTP client Postman.

- Course Outcomes**  
On completion of the course, student will be able to–
1. Learn the fundamentals of Blockchain Technology.
  2. Learn Blockchain programming
  3. Basic knowledge of Smart Contracts and how they function.

**Course Contents**

<b>Chapter 1</b>	<b>Introduction to Blockchain</b>	<b>7 Lect</b>
------------------	-----------------------------------	---------------

- Foundational Computing Concepts (Client-Server systems vs Peer to Peer Systems)
- Evolution of Blockchain
- Blockchain Vs Database
- Essentials of Blockchain (Blockchain generations, types of blockchain, benefits and challenges of blockchain usage)
- Types of Networks
- Layered Architecture of Blockchain Ecosystem
- Components of blockchain
- Cryptography (private and public keys, Hashing & Digital Signature)
- Consensus Mechanisms
- Cryptocurrency, Digital Currency Bitcoin and Ethereum
- Smart Contracts
- Blockchain use cases

<b>Chapter 2</b>	<b>How Blockchain Works?</b>	<b>5 Lect</b>
------------------	------------------------------	---------------

- Understanding SHA256 Hash
- Immutable Ledger
- Distributed P2P Network
- How Mining Works? (The NONCE and Cryptographic Puzzle)
- Byzantine Fault Tolerance
- Consensus Protocols: Proof of Work, Proof of State, Défense Against Attackers, Competing Chains
- Blockchain Demo



<b>Chapter 3</b>	<b>Smart Contracts</b>	<b>6 Lect</b>
<ul style="list-style-type: none"> <li>• Ethereum Network</li> <li>• What is a Smart Contract?</li> <li>• Ethereum Virtual Machine, Ether, Gas</li> <li>• DApps</li> <li>• Decentralized Autonomous Organizations (DAO)</li> <li>• Hard and Soft Forks</li> <li>• Initial Coin Offerings</li> <li>• Demo of Smart Contracts</li> </ul>		
<b>Demonstration</b>	<b>Programming Assignments:</b>	<b>18 Lect</b>
<p><b>Out of 36 lectures, 18 are assigned for demonstration. Teacher should give demonstration of various programs mentioned below in the classroom or in the laboratory as per their convenience.</b></p> <p><b>Assignment 1</b> –Demonstration of Blockchain  <a href="https://andersbrownworth.com/blockchain">https://andersbrownworth.com/blockchain</a></p> <p><b>Assignment 2</b> – Installation of Ganache, Flask and Postman</p> <p><b>Assignment 3</b> –Write a Simple Python program to create a Block class that contains index, timestamp, and previous hash. Connect the blocks to create a Blockchain.</p> <p><b>Assignment 4</b> –<b>Demo of Remix-Ethereum IDE</b> <a href="https://remix.ethereum.org">https://remix.ethereum.org</a> and Test Networks</p> <p><b>Assignment 5</b>–1. Write a Simple Smart Contract for Bank with withdraw and deposit functionality.</p> <p><b>Assignment 6</b> – 2. Write a Smart Contract for storing and retrieving information of Degree Certificates.</p>		
<b>Reference Books:</b>		
<b>Textbook:</b>		
<ol style="list-style-type: none"> <li>1. <b>Beginning Blockchain : A Beginner’s Guide to Building Blockchain Solutions By Bikramaditya Singhal, Gautam Dhameja, Priyansu Sekhar Panda, Apress Media</b></li> </ol>		
<b>Reference Books:</b>		
<ol style="list-style-type: none"> <li>2. Mastering Blockchain by Imran Bashir, Third Edition, Packt Publication</li> <li>3. Waterhole, The Science of the Blockchain</li> <li>4. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System</li> <li>5. Mastering Ethereum: Building Smart Contracts and DAPPS, by Andreas Antonopoulos, Dr. Gavid Wood, Oreilly Publication</li> </ol>		
<b>Reference Web Links</b>		
<ol style="list-style-type: none"> <li>1. <a href="https://www.investopedia.com/terms/b/blockchain.asp">https://www.investopedia.com/terms/b/blockchain.asp</a></li> </ol>		

**Savitribai Phule Pune University**  
**T.Y.B.Sc. (Computer Science) - Sem – VI**  
**Course Type: DSEC - IV                      Course Code: CS - 361**  
**Course Title : Operating Systems-II**

Teaching Scheme: 03 Lect / week	No. of Credits: 2	Examination Scheme: IE : 15 marks UE: 35 marks
------------------------------------	----------------------	--

**Prerequisites**  
Concepts of Operating System, Processes and Threads Scheduling, Synchronization

- Course Objectives:**
1. To understand the issue of Deadlocks in Process management.
  2. To understand the concept of File system management & disk scheduling
  3. To study the concept of distributed and mobile operating systems

- Course Outcomes: After completion of this course students will be able to understand the concept of**
1. Management of deadlocks and File System by operating system
  2. Scheduling storage or disk for processes
  3. Distributed Operating System and its architecture and the extended features in mobile OS.

**Course Contents**

<b>Chapter 1</b>	<b>Process Deadlocks</b>	<b>7 lectures</b>
	<ul style="list-style-type: none"> <li>• System model</li> <li>• Deadlock Characterization – Necessary conditions, Resource allocation graph</li> <li>• Deadlock Methods- Prevention and Deadlock Avoidance - Safe state, Resource allocation graph algorithm, Banker’s Algorithm</li> <li>• Deadlock Detection</li> <li>• Recovery from Deadlock – Process termination, Resource preemption</li> </ul>	
<b>Chapter 2</b>	<b>File system Management</b>	<b>6 lectures</b>
	<ul style="list-style-type: none"> <li>• File concept , File attributes, File operations</li> <li>• Access Methods – Sequential, Direct, Other access methods</li> <li>• Directory overview, Single level directory, Two level directory, Tree structure directory, Acyclic graph directory, General graph directory</li> <li>• Allocation Methods – Contiguous allocation, Linked allocation, Indexed allocation</li> <li>• Free Space Management – Bit vector, Linked list, Grouping, Counting, Space maps</li> </ul>	
<b>Chapter 3</b>	<b>Disk scheduling</b>	<b>4 lectures</b>
	<ul style="list-style-type: none"> <li>• Overview, Disk Structure</li> <li>• Disk Scheduling, FCFS Scheduling, SSTF Scheduling, Scan Scheduling-Scan Scheduling, Look Scheduling , Disk Management</li> </ul>	
<b>Chapter 4</b>	<b>Introduction to Distributed operating systems &amp; Architecture</b>	<b>11 lectures</b>
	<ul style="list-style-type: none"> <li>• What is a distributed system, Design goals</li> <li>• Types of distributed systems</li> <li>• Architectural styles : Layered architectures , Object-based architectures, Resource-centered architectures</li> <li>• System architecture – Centralized organization, Decentralized organizations, peer-to-peer systems, Hybrid architectures.</li> </ul>	

<ul style="list-style-type: none"> <li>• Example architectures : Network file system(NFS), Web-based distributed systems</li> </ul>		
<b>Chapter 5</b>	<b>Mobile Operating Systems</b>	<b>7 lectures</b>
<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Features</li> <li>• Special Constraints and Requirements of Mobile Operating System</li> <li>• Special Service Requirements</li> <li>• ARM &amp; Intel architectures – Power management</li> <li>• Mobile OS architectures – Underlying OS, kernel structure &amp; native level programming, Runtime issues, Approaches to power management</li> <li>• Commercial Mobile Operating Systems - Windows Mobile, iPhone OS (iOS), Android</li> <li>• A Comparative Study of Mobile Operating Systems (Palm OS, Android, Symbian OS, Blackberry OS, Apple iOS)</li> </ul>		
<b>Reference Books:</b>		
<ol style="list-style-type: none"> <li>1) Advanced Concepts in Operating Systems, M Singhal and NG Shivaratri, Tata McGraw Hill Inc, 2001 (Text Book)</li> <li>2) Operating System Concepts, Avi Silberschatz, Peter Galvin, Greg Gagne, Student Edition, Wiley Asia</li> <li>3) Operating Systems: Internals and Design Principles, William Stallings, Prentice Hall of India.</li> <li>4) Distributed Operating Systems Concepts and Design, Pradeep K. Sinha, PHI</li> <li>5) Prasant Kumar Pattnaik, Rajib Mall, “Fundamentals of Mobile Computing”, PHI Learning Pvt.Ltd, New Delhi – 2012.</li> <li>6) A. Tannenbum, Herbert Bos, “Modern Operating systems”, Pearson Publication, 4<sup>th</sup> Edition</li> <li>7) A. Tannenbum, Maarten van Steen, “Distributed systems”, 3<sup>rd</sup> Edition</li> <li>8) Source wikipedia, Mobile operating systems, General books, LLC,2010</li> </ol>		



Usability and accessibility testing Configuration, compatibility testing		
<b>Chapter 4</b>	<b>Testing Web Applications</b>	<b>6 lectures</b>
Dimension of Quality, Error within a WebApp Environment Testing Strategy for WebApp Test Planning The Testing Process –an overview		
<b>Chapter 5</b>	<b>Agile Testing</b>	<b>5 lectures</b>
Agile Testing, Difference between Traditional and Agile testing, Agile principles and values, Agile Testing Quadrants, Automated Tests.		
<b>Reference Books:</b>		
<ol style="list-style-type: none"> <li>1. Software Engineering – A Practitioners Approach, Roger S. Pressman, 7<sup>th</sup> Edition, Tata McGraw Hill, 20</li> <li>2. Effective Methods of Software Testing, William E Perry, 3<sup>rd</sup> Edition, Wiley Publishing Inc</li> <li>3. Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing, Rex Black, Microsoft Press, 1999</li> <li>4. Agile Testing: A Practical Guide for Testers and Agile Teams, Lisa Crispin and Janet Gregory, 1<sup>st</sup> Edition, Addison-Wesley Professional, 2008</li> <li>5. Software Testing Principles and Practices By Srinivasan Desikan, Gopaldaswamy Ramesh, Pearson</li> </ol>		



JS Strings and JS String methods		
JS popup boxes(alert, confirm, prompt). Jquery library , Including jquery library in page Jquery selector , DOM manipulation using jquery		
<b>Chapter 4</b>	AJAX	<b>6 Lect</b>
Introduction of AJAX AJAX web application model AJAX –PHP framework Performing AJAX validation Handling XML data using php and AJAX Connecting database using php and AJAX		
<b>Chapter 5</b>	PHP framework CodeIgniter	<b>8 Lect</b>
CodeIgniter - Overview, Installing CodeIgnite Application Architecture MVC Framework , Basic concept of CodeIgniter, Libraries Working with databases Load external JS and CSS page & redirecting from controller , Adding JS and CSS , Page redirection. Loading dynamic data on page & session management, cookies management		
<b>Reference Books:</b>		
<ol style="list-style-type: none"> <li>1. <b>Programming PHP By Rasmus Lerdorf and Kevin Tatroe O'Reilly publication</b></li> <li>2. <b>Beginning PHP 5, Wrox publication</b></li> <li>3. <b>AJAX Black Book Kogent solution</b></li> <li>4. <b>Mastering PHP BPB Publication</b></li> <li>5. <b>Professional Codeigniter By Thomas Myer ,Wrox Publication,</b></li> <li>6. <b>Codeihniter 2 CookBook By Rob Foster ,PACKT Publication ,</b></li> <li>7. <b>JQuery CookBook, O'reilly Publication.</b></li> </ol>		
Ref. Links:		
<ol style="list-style-type: none"> <li>1. <a href="http://www.php.net.in">www.php.net.in</a></li> <li>2. <a href="http://www.W3schools.com">www.W3schools.com</a></li> <li>3. <a href="https://www.tutorialspoint.com/codeigniter/index.htm">https://www.tutorialspoint.com/codeigniter/index.htm</a></li> <li>4. <a href="https://api.jquery.com/">https://api.jquery.com/</a></li> <li>5. <a href="http://codeigniter.com/docs">http://codeigniter.com/docs</a></li> </ol>		

**Savitribai Phule Pune University**  
**T.Y.B.Sc. (Computer Science) – Sem VI**  
**Course Type: DSEC – V**                      **Course Code: CS - 364**  
**Course Title : Data Analytics**

Teaching Scheme 03 lectures / week	No. of Credits 2	Examination Scheme IE : 15 marks UE: 35 marks
---------------------------------------	---------------------	---

**Prerequisites**

- Basic of mathematics and statistics
- Basic programming Knowledge of python
- Knowledge of databases

**Course Objectives**

- Deploy the Data Analytics Lifecycle to address data analytics projects.
- Develop in depth understanding of the key technologies in data analytics.
- Apply appropriate analytic techniques and tools to analyze data, create models, and identify insights that can lead to actionable results.

**Course Outcomes**

On completion of the course, student will be able to–

- Use appropriate models of analysis, assess the quality of input, and derive insight from results.
- Analyze data, choose relevant models and algorithms for respective applications
- Understand different data mining techniques like classification, prediction, clustering and association rule mining
- Apply modeling and data analysis techniques to the solution of real world business problems

**Course Contents**

<b>Chapter 1</b>	<b>Introduction to Data Analytics</b>	<b>6 lectures</b>
------------------	---------------------------------------	-------------------

Concept of data analytics  
Data analysis vs Data analytics  
Types of analytics  
    Diagnostic Analytics, Predictive Analytics , Prescriptive Analytics, Exploratory Analysis, Mechanistic Analysis  
Mathematical models - Concept  
Model evaluation: metrics for evaluating classifiers - Class imbalance - AUC, ROC (Receiver-Operator Characteristic) curves, Evaluating value prediction models

<b>Chapter 2</b>	<b>Machine Learning Overview</b>	<b>6 Lectures</b>
------------------	----------------------------------	-------------------

Introduction to Machine Learning, deep learning, Artificial intelligence  
Applications for machine learning in data science  
The modeling process  
    Engineering features and selecting a model, Training the model, Validating the model, Predicting new observations  
Types of machine learning  
    Supervised learning, Unsupervised learning, Semi-supervised learning, ensemble techniques  
Regression models  
    Linear Regression



Polynomial Regression Logistic Regression 2.6. Concept of classification, clustering and reinforcement learning.		
<b>Chapter 3</b>	<b>Mining Frequent Patterns, Associations, and Correlations</b>	<b>12 lectures</b>
What kind of patterns can be mined Class/Concept Description: Characterization and Discrimination, Mining Frequent Patterns, Associations, and Correlations, Classification and Regression for Predictive Analysis, Cluster Analysis, Outlier Analysis Mining frequent patterns - Market Basket Analysis. Frequent Itemsets, Closed Itemsets, and Association Rules Frequent Itemset Mining Methods Apriori Algorithm Generating Association Rules from Frequent Itemsets Improving efficiency of apriori algorithm Frequent pattern growth (FP-growth) algorithm		
<b>Chapter 4</b>	<b>Social Media and Text Analytics</b>	<b>12 lectures</b>
Overview of social media analytics Social Media Analytics Process, Seven layers of social media analytics, accessing social media data Key social media analytics methods Social network analysis Link prediction, Community detection, Influence maximization, Expert finding, Prediction of trust and distrust among individuals Introduction to Natural Language Processing Text Analytics : Tokenization, Bag of words, Word weighting : TF-IDF, n-Grams, stop words, Stemming and lemmatization, synonyms and parts of speech tagging Sentiment Analysis Document or text summarization Trend analytics Challenges to social media analytics		
<b>Reference Books:</b>		
1) Data Science Fundamentals and Practical Approaches, Gypsy Nandi, Rupam Sharma, BPB Publications, 2020. 2) The Data Science Handbook, Field Cady, John Wiley & Sons, Inc, 2017 3) Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber, Jian Pei, Morgan Kaufmann, Third Edition, 2012. 4) A Hands-On Introduction to Data Science, Chirag Shah, University of Washington Cambridge University Press 5) The Data Science Design Manual, Steven S. Skiena, Springer, 2017 6) Introducing data science: big data, machine learning, and more, using Python tools, Cielen D., Meysman A. D., & Ali M., Manning Publications Co., 2016		

**Savitribai Phule Pune University**  
**T.Y.B.Sc. (Computer Science) - Sem – VI**  
**Course Type: DSEC – VI                      Course Code : CS - 365**  
**Course Title : Object Oriented Programming using Java – II**

Teaching Scheme 03 Lect / week	No. of Credits 2	Examination Scheme IE : 15 marks UE: 35 marks
-----------------------------------	---------------------	---

**Prerequisites**

- Knowledge of Core Java (CS – 355)

**Course Objectives**

- To learn database programming using Java
- To study web development concept using Servlet and JSP
- To develop a game application using multithreading
- To learn socket programming concept

**Course Outcomes**

On completion of the course, student will be able to–

- To access open database through Java programs using Java Data Base Connectivity (JDBC) and develop the application.
- Understand and Create dynamic web pages, using Servlets and JSP.
- Work with basics of framework to develop secure web applications.

**Course Contents**

<b>Chapter 1</b>	<b>Collections</b>	<b>6 Lect</b>
------------------	--------------------	---------------

Introduction to the Collection framework  
List - ArrayList, LinkedList  
Set - HashSet, TreeSet,  
Map - HashMap and TreeMap  
Interfaces such as Comparator, Iterator, ListIterator, Enumeration

<b>Chapter 2</b>	<b>Multithreading</b>	<b>6 Lect</b>
------------------	-----------------------	---------------

What are threads?  
Life cycle of thread  
Creating threads - Thread class , Runnable interface  
Thread priorities  
Running multiple threads  
Synchronization and interthread communication

<b>Chapter 3</b>	<b>Database Programming</b>	<b>6 Lect</b>
------------------	-----------------------------	---------------

The design of jdbc  
Types of drivers  
Executing sql statements, query execution  
Scrollable and updatable Resultset

<b>Chapter 4</b>	<b>Servlets and JSP</b>	<b>12 Lect</b>
------------------	-------------------------	----------------

Introduction to Servlet and Hierarchy of Servlet  
Life cycle of servlet  
Handling get and post request (HTTP)  
Handling data from HTML to servlet  
Retrieving data from database to servlet

Session tracking – User Authorization, URL rewriting, Hidden form fields, Cookies and HttpSession

Introduction to JSP, Life cycle of JSP

Implicit Objects

Scripting elements - Declarations, Expressions, Scriptlets, Comments

JSP Directives - Page Directive, include directive

Mixing Scriptlets and HTML

JSP Actions - jsp:forward , jsp:include, jsp:useBean, jsp:setProperty and jsp:getProperty

**Chapter 5**

**Spring Framework**

**6 Lect**

Introduction of Spring framework

Spring Modules / Architecture

Spring Applications

Spring MVC

Spring MVC Forms, Validation

**Reference Books:**

R1. Complete reference Java by Herbert Schildt(5th edition)

R2. Java 2 programming black books, Steven Horlzner

R3. Programming with Java , A primer ,Forth edition , By E. Balagurusamy

R4. Core Java Volume-I-Fundamentals, Eighth Edition, Cay S. Horstmann, Gary Cornell, Prentice Hall, Sun Microsystems Press

R5. Core Java Volume-II-Advanced Features, Eighth Edition, Cay S. Horstmann, Gary Cornell, Prentice Hall, Sun Microsystems Press

R6. Getting started with Spring Framework: covers Spring 5 by J Sharma and Ashish Sarin

R7. Spring 4 for Developing Enterprise Applications: An End-to-End Approach by Henry H. Liu

<p style="text-align: center;"><b>Savitribai Phule Pune University</b>  <b>T.Y.B.Sc. (Computer Science) - Sem - VI</b>  <b>Course Type: DSEC - VI</b>                      <b>Course Code: CS - 366</b>  <b>Course Title: Compiler Construction</b></p>		
Teaching Scheme 3 Lect / week	No. of Credits 2	Examination Scheme IE: 15 marks UE: 35 marks
<p><b>Prerequisites</b></p> <ul style="list-style-type: none"> <li>• Knowledge of Automata Theory and Languages.</li> </ul>		
<p><b>Course Objectives</b></p> <ul style="list-style-type: none"> <li>• To understand design issues of a lexical analyzer and use of LEX tool.</li> <li>• To understand design issues of a parser and use of YACC tool.</li> <li>• To understand and design code generation and optimization techniques.</li> </ul>		
<p><b>Course Outcomes</b></p> <p>On completion of the course, student will be able to–</p> <ul style="list-style-type: none"> <li>• Understand the process of scanning and parsing of source code.</li> <li>• Learn the conversion code written in source language to machine language.</li> <li>• Understand tools like LEX and YACC.</li> </ul>		
<b>Course Contents</b>		
<b>Chapter 1</b>	<b>Introduction</b>	<b>4 Lect</b>
<p>Definition of Compiler, Aspects of compilation.  The structure of Compiler.  Phases of Compiler – Lexical Analysis, Syntax Analysis, Semantic Analysis,  Intermediate Code generation, code optimization, code generation.  Error Handling.  Introduction to one pass &amp; Multipass compilers, cross compiler, Bootstrapping.</p>		
<b>Chapter 2</b>	<b>Lexical Analysis (Scanner)</b>	<b>4 Lect</b>
<p>Review of Finite automata as a lexical analyzer,  Applications of Regular Expressions and Finite Automata (lexical analyzer,  searching using RE), Input buffering, Recognition of tokens.  LEX: A Lexical analyzer generator (Simple Lex Program)</p>		
<b>Chapter 3</b>	<b>Syntax Analysis (Parser)</b>	<b>14 Lect</b>
<p>Definition, Types of Parsers  Top-Down Parser –  Top-Down Parsing with Backtracking: Method &amp; Problems  Drawbacks of Top-Down parsing with backtracking, 3.2.3 Elimination of Left  Recursion (direct &amp; indirect) 3.2.4 Need for Left Factoring &amp; examples  Recursive Descent Parsing: Definition  Implementation of Recursive Descent Parser Using Recursive Procedures  3.4 Predictive [LL (1)] Parser (Definition, Model)  3.4.1 Implementation of Predictive Parser [LL (1)]  3.4.2 FIRST &amp; FOLLOW</p>		

<p>Construction of LL (1) Parsing Table          Parsing of a String using LL (1) Table.</p> <p>Bottom-Up Parsers</p> <p>Operator Precedence Parser -Basic Concepts</p> <p>Operator Precedence Relations form Associativity &amp; Precedence</p> <p>Operator Precedence Grammar</p> <p>Algorithm for LEADING &amp; TRAILING (with ex.)</p> <p>Algorithm for Operator Precedence Parsing (with ex.)</p> <p>Precedence Functions</p> <p>Shift Reduce Parser</p> <p>Reduction, Handle, Handle Pruning</p> <p>Stack Implementation of Shift Reduce Parser (with examples)</p> <p>LR Parser: Model, Types [SLR (1), Canonical LR, LALR]-Method &amp; examples.</p> <p>YACC (from Book 3) –program sections, simple YACC program for expression evaluation</p>		
<b>Chapter 4</b>	<b>Syntax Directed Definition</b>	<b>7 Lect</b>
<p>Syntax Directed Definitions (SDD)</p> <p>Inherited &amp; Synthesized Attributes</p> <p>Evaluating an SDD at the nodes of a Parse Tree, Example</p> <p>Evaluation Orders for SDD's</p> <p>Dependency Graph</p> <p>Ordering the Evaluation of Attributes</p> <p>S-Attributed Definition</p> <p>L-Attributed Definition</p> <p>Application of SDT</p> <p>Construction of syntax trees,          The Structure of a Type</p> <p>4. 4 Translation Schemes</p> <p>4.4.1 Definition, Postfix Translation Scheme</p>		
<b>Chapter 5</b>	<b>Code Generation and Optimization</b>	<b>7 Lect</b>
<p>Compilation of expression –</p> <p>Concepts of operand descriptors and register descriptors with example.</p> <p>Intermediate code for expressions – postfix notations,          Triples, Quadruples and Expression trees.</p> <p>Code Optimization – Optimizing transformations – compile time evaluation, elimination of common sub expressions, dead code elimination, frequency reduction, strength reduction.</p> <p>Three address code</p> <p>DAG for Three address code</p> <p>The Value-number method for constructing DAG's.</p> <p>Definition of basic block, Basic blocks, and flow graphs</p> <p>Directed acyclic graph (DAG) representation of basic block.</p> <p>Issues in design of code generator.</p>		

**Reference Books**

1. Compilers: Principles, Techniques, and Tools, Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, 2004
2. Principles of Compiler Design By: Alfred V. Aho, Jeffrey D. Ullman, Narosa Publication House, 2002
3. LEX & YACC, 2<sup>nd</sup> edition, O'reilly Publication, 2012

**Savitribai Phule Pune University**  
**T.Y.B.Sc. (Computer Science) - Sem – VI**  
**Course Type: DSEC- IV                      Course Code: CS - 367**  
**Course Title : Practical Course based on CS - 361**

Teaching Scheme:  
5 Lect/ week

No. of Credits:  
2

Examination Scheme:  
IE : 15 marks  
UE: 35 marks

**Course Objectives:**

1. To implement Banker's algorithm for Deadlocks in Process management.
2. To simulate File system management
3. To study and implement various algorithms of disk scheduling

**Course Outcomes: After completion of this course students will be able to understand the concept of**

1. Management of deadlocks by operating system
2. File System management
3. Disk space management and scheduling for processes

**Guidelines:**

1. Operating system platform – Linux
2. Programming language - C

**List of Assignments:**

- Simulation of Banker's algorithm of deadlock avoidance in processes of operating system ( 3 slots)
- Simulation of File Allocation methods and free space management in storage - Contiguous allocation, Linked allocation, Indexed allocation (4 slots)
- Simulation of Disk Scheduling algorithms – FCFS, SSTF, Scan, Look (2 slots)
- Assignment based on distributed and mobile OS ( 3 slots)





**SavitribaiPhule Pune University**  
**T.Y.B.Sc. (Computer Science) - Sem – VI**  
**Course Type: DSEC - VI                      Course Code: CS - 369**  
**Course Title : Practical Course based on CS - 365**

Teaching Scheme 5 Lect/ week Batch Size : 12	No. of Credits 2	Examination Scheme IE : 15 marks UE: 35 marks
--	---------------------	---

**Course Objectives:**

1. Covers the complete scope of the syllabus.
2. Bringing uniformity in the way course is conducted across different colleges.
3. Continuous assessment of the students.
4. Advanced Java is designed to develop web based, network centric, Enterprise level applications

**Course Outcomes:**

1. To Learn database Programming using Java
2. Understand and Create dynamic web pages using Servlets and JSP.
3. Work with basics of framework to develop secure web applications

**Guidelines:**

**Operating Environment :**

- **Operating system:** Linux
- **Editor:** Anylinux based editor like vi, gedit and Use of IDE – Eclipse etc.
- **Compiler :** javac
- **Database :** postgresql

**Submission :**

Each assignment will be assessed on a scale of 0 to 5 as indicated below.

- |                     |   |
|---------------------|---|
| ▪ Not done          | 0 |
| ▪ Incomplete        | 1 |
| ▪ Late Complete     | 2 |
| ▪ Needs improvement | 3 |
| ▪ Complete          | 4 |
| ▪ Well Done         | 5 |

**Assessment :**

Easy : All exercises are compulsory.  
Medium : All exercises are compulsory.

**List of Assignments:**

**Assignment 1 : Database Programming [Slot-2]**

Study the Collection framework in java.  
To Implement various Interfaces and classes through algorithms.  
To Demonstrate Cursor Objects (Enumeration, Iterator, ListIterator, Comparator)

**Assignment 2 : Multithreading [Slot-2]**

To create and use threads in java.  
To demonstrate multithreading using Thread Synchronization, Inter-thread .  
Communication, Thread Priorities.

**Assignment 3 : Database Programming [Slot-2]**

To communicate with a database using java.  
To execute queries on tables.  
To obtain information about the database and tables.

**Assignment 4 : Servlets [Slot-2]**

To understand server-side programming.  
Simple steps to create and execute servlets.  
How to pass parameters using doGet and doPost methods.  
Handling data from HTML to servlet .  
How to connect servlet to a database .  
Use of various session tracking methods like Cookies.

**Assignment 5 : Java Server Pages [Slot-2]**

JSP life-cycle.  
Use of JSP implicit objects.  
JSP Directives.  
Use of Scripting Elements.  
To understand actionstags in JSP.  
Understanding flow of JSP custom tags.

**Assignment 6 : Spring Framework [Slot-2]**

To create and understand the steps to develop Spring application.

**Savitribai Phule Pune University**  
**T.Y.B.Sc. (Computer Science) - Sem – VI**  
**Course Type: SECC - III                      Course Code: CS - 3610**  
**Course Title: Software Testing Tools**

Teaching Scheme: 03 Lect / week	No. of Credits: 2	Examination Scheme: IE:15 marks UE: 35 marks
------------------------------------	----------------------	--

**Prerequisites**

- Basic knowledge of algorithms, problem solving, expected inputs/outputs
- Knowledge of C and Java Programming Language, compilation, debugging

**Course Objectives:**

- To provide the knowledge of software testing methods and strategies.
- To understand how testing methods can be used as an effective tool in quality assurance of software.
- To provide skills to design test case plan for testing software.
- 4.To provide knowledge of latest testing tools

**Course Outcomes:**

- To understand various software testing methods and strategies.
- To understand a variety of software metrics and identify defects and managing those defects for improvement in quality for given software.
- To design test cases and test plans, review reports of testing for qualitative software.
- 4. To understand latest testing tools used in the software industries.

**Course Contents**

<b>Chapter 1</b>	<b>Introduction to Test case design</b>	<b>4 lectures</b>
How to identify errors, bugs in the given application. Design entry and exit criteria for test case, design test cases in excel. Describe feature of a testing method used.		
<b>Chapter 2</b>	<b>Test cases for simple programs</b>	<b>4 lectures</b>
Write simple programs make use of loops and control structures. Write Test Cases for above programs.		
<b>Chapter 3</b>	<b>Test cases and Test plan</b>	<b>4 lectures</b>
Write Test Plan for given application with resources required. Write Test case for given application. Prepare Test report for test cases executed.		
<b>Chapter 4</b>	<b>Defect Report</b>	<b>3 lectures</b>
Defect Life Cycle Classification of Defect Write Defect Report		
<b>Chapter 5</b>	<b>Testing Tools</b>	<b>3 lectures</b>
How to make use of Automation Tools Types of Testing Tools		
<b>Demonstration</b>	<b>Programming Assignments</b>	<b>18 Lect</b>

**Out of 36 lectures, 18 are assigned for demonstration. Teacher should give demonstration of various assignments based on above theory topics in the classroom or in the laboratory as per their convenience. Demonstration of any open source testing tool should be given.**

Programming assignments should be done individually by the student in their respective login from the list given in Labbook. The code/ documentation should be uploaded on either the local server, Moodle, Github or any LMS.

**Reference Books:**

1. Software Engineering – A Practitioners Approach, Roger S. Pressman, 7<sup>th</sup> Edition, Tata McGraw Hill, 20
2. Effective Methods of Software Testing, William E Perry, 3<sup>rd</sup> Edition, Wiley Publishing Inc
3. Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing, Rex Black, Microsoft Press, 1999
4. Software Testing Principles and Practices by Srinivasan Desikan, Gopalswamy Ramesh, Pearson.

**Savitribai Phule Pune University**  
**T.Y.B.Sc. (Computer Science) - Sem - VI**  
**Course Type: SECC - IV      Course Code: CS - 3611**  
**Course Title : Project**

Teaching Scheme 03 Lect/ week/Batch Batch Size : 20	No. of Credits 2	Examination Scheme IE : 15 marks UE: 35 marks
---	---------------------	---

**Project Guidelines:**

- Students should work in a team of minimum 3 and maximum 4 students.
- Students can choose a project topic and implement the same using any language/technology covered in the curriculum so far. The operating environment must be linux.
- The student group will work independently throughout the project work including: problem identification, information searching, literature study, design and analysis, implementation, testing, and the final reporting.
- Project guide must conduct project presentations (minimum 2) to monitor the progress of the project groups.
- At the end of the project, the group should prepare a report which should conform to international academic standards. The report should follow the style in academic journals and books, with clear elements such as: abstract, background, aim, design and implementation, testing, conclusion and full references, Tables and figures should be numbered and referenced to in the report.
- The final project presentation with demonstration (UE) will be evaluated by the project guide (appointed by the college) and one external examiner (appointed by the University).

**Recommended Documentation contents:**

**Abstract**

**Introduction**

- motivation
- problem statement
- purpose/objective and goals
- literature survey
- project scope and limitations

**System analysis**

- Existing systems
- scope and limitations of existing systems
- project perspective, features
- stakeholders
- Requirement analysis - Functional requirements, performance requirements, security requirements etc.

**System Design**

- Design constraints
- System Model: Using OOSE
- Data Model
- User interfaces

**Implementation details**

- Software/hardware specifications

**Outputs and Reports Testing**

- Test Plan, Black Box Testing or Data Validation Test Cases, White Box Testing or Functional Validation Test cases and results

**Conclusion and Recommendations**

**Future Scope**

**Bibliography and References**

**Project Related Assignments**

**Guidelines:**

- The project assignments are a compulsory part of the project course and should be carried out by each project group.
- Project assignments are to be given by the guide for continuous internal evaluation.
- The project assignments are to be allotted to each group separately by the project guide on the basis of the implementation technology. A suggested list of assignments is given below.
  1. Project Time management: plan (schedule table), Gantt chart, Roles and responsibilities, data collection, Implementation
  2. Simple assignments to evaluate choice of technology
  3. Assignments on UI elements in chosen technology
  4. Assignments on User interfaces in the project
  5. Assignments on event handling in chosen technology
  6. Assignments on Data handling in chosen technology
  7. Online and offline connectivity
  8. Report generation
  9. Deployment considerations
  10. Test cases
- Each student within the group must work actively and contribute to the assignments, project work and report writing.

**Evaluation guidelines:**

IA (15 marks)			UE (35 marks)		
First presentation	Second presentation	Assignments	Project Logic/ Presentation	Assignments and Project Documentation	Viva
05	05	05	20	10	05