UNIVERSITY OF MUMBAI

Syllabus for Semester III and IV
Program: M.Sc.
Course: Information Technology

(Credit Based Semester and Grading System with effect from the academic year 2013–2014)
M.Sc.
Information Technology

Semester III

<table>
<thead>
<tr>
<th>Paper code</th>
<th>Paper Nomenclature</th>
<th>Lectures</th>
<th>Credit</th>
<th>Practical Paper</th>
<th>Hrs</th>
<th>Credit</th>
<th>Total Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSIT301</td>
<td>Software Testing</td>
<td>60</td>
<td>04</td>
<td>PSITP9</td>
<td>60</td>
<td>02</td>
<td>06</td>
</tr>
<tr>
<td>PSIT302</td>
<td>Artificial Intelligence</td>
<td>60</td>
<td>04</td>
<td>PSITP10</td>
<td>60</td>
<td>02</td>
<td>06</td>
</tr>
<tr>
<td>PSIT303</td>
<td>Parallel Processing</td>
<td>60</td>
<td>04</td>
<td>PSITP11</td>
<td>60</td>
<td>02</td>
<td>06</td>
</tr>
<tr>
<td>PSIT304</td>
<td>Multimedia systems and convergence of technologies</td>
<td>60</td>
<td>04</td>
<td>PSITP12</td>
<td>60</td>
<td>02</td>
<td>06</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24</td>
</tr>
</tbody>
</table>

Semester IV

<table>
<thead>
<tr>
<th>Paper code</th>
<th>Paper Nomenclature</th>
<th>Lectures</th>
<th>Credit</th>
<th>Practical Paper</th>
<th>Hrs</th>
<th>Credit</th>
<th>Total Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSIT401</td>
<td>Information Security</td>
<td>60</td>
<td>04</td>
<td>PSITP13</td>
<td>60</td>
<td>02</td>
<td>06</td>
</tr>
<tr>
<td>PSIT402</td>
<td>Robotics</td>
<td>60</td>
<td>04</td>
<td>PSITP14</td>
<td>60</td>
<td>02</td>
<td>06</td>
</tr>
<tr>
<td>PSIT403</td>
<td>Distributed Computing</td>
<td>60</td>
<td>04</td>
<td>PSITP15</td>
<td>60</td>
<td>02</td>
<td>06</td>
</tr>
<tr>
<td>PSIT404</td>
<td>JAVA Technology</td>
<td>60</td>
<td>04</td>
<td>PSITP16</td>
<td>60</td>
<td>02</td>
<td>06</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24</td>
</tr>
</tbody>
</table>

Total credits for M.Sc. Sem III and IV = (Sem III- 24 and sem IV-24) =48

Evaluation: The students will be evaluated externally. The external evaluation will be done by the committee appointed by the University norms. Standard of passing and scale as per the university norms.

Information Technology Syllabus
Restructured for Credit Based and Grading System

SEM: III

Paper I: Software Testing
Paper II: Artificial Intelligence
Paper III: Parallel Processing
Paper IV: Multimedia systems and convergence of technologies

SEM: IV

Paper IV: Information Security
Paper V: Robotics
Paper VI: Distributed Computing
Paper VIII: Java Technology
### SEMESTER III

**Course 1: SOFTWARE TESTING**  
**PSIT301**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Unit</th>
<th>Description</th>
<th>Credits</th>
</tr>
</thead>
</table>
| PSIT301     | I    | **Introduction:** Defect, Defect Vs failures, Process problems and defect rates, The business perspective for testing  
**Building a Software Testing Strategy:** Computer system strategic risk, Economics of testing, Common computer problems, Economics of SDLC testing, Testing- an organizational issue, Establishing a testing policy, Structured approach to testing, Test strategy, Testing methodology | 4 |
|             | II   | **Establishing a Software Testing Methodology:** Introduction, Verification and validation, Functional and structural testing, Workbench concept, Considerations in developing testing methodologies  
**Determining Software Testing Techniques:** Testing techniques/tool selection process, Selecting techniques/tools, Structural system testing techniques, Functional system testing techniques, Unit testing techniques, Functional testing and analysis |  |
|             | III  | **Selecting and Installing Software Testing Tools:** Testing tools-Hammers of testing, Selecting and using the test tools, Appointing managers for testing tools  
**Software Testing Process:** Cost of computer testing, Life cycle testing concept, Verification and validation in the software development process, Software testing process, Workbench skills |  |
|             | IV   | **Software Testing Process:** Access Project Management Development Estimate and Status, Test Plan, Requirements Phase Testing, Design Phase Testing, Program Phase Testing, Execute Test and Record Results, Acceptance Test, Report Test Result, Testing Software Installation, Test Software Change, Evaluate Test Effectiveness |  |
|             | V    | **Testing Specialized Systems and Applications:** Client/Server systems, RAD, System documentation, Web based systems, Off-the-self software, Multi platform environment, Security, Data Warehouse  
**Building Test Document:** Uses, Types, Responsibility, Storage, Test plan documentation, Test analysis report documentation |  |

**Text Books**


**References :**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Unit</th>
<th>Description</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSIT302</td>
<td>I</td>
<td><strong>AI and Internal Representation</strong>: Artificial Intelligence and the World, Representation in AI, Properties of Internal Representation, The Predicate Calculus, Predicates and Arguments, Connectives Variables and Quantification, How to Use the Predicate Calculus, Other Kinds of Inference Indexing, Pointers and Alternative Notations, Indexing, The Isa Hierarchy, Slot-Assertion Notation, Frame Notation</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td><strong>Lisps</strong>: Lisps, Typing at Lisp, Defining Programs, Basic Flow of Control in Lisp, Lisp Style, Atoms and Lists, Basic Debugging, Building Up List Structure, More on Predicates, Properties, Pointers, Cell Notation and the Internals (Almost) of Lisp, Destructive Modification of Lists, The for Function, Recursion, Scope of Variables, Input/Output, Macros</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>III</td>
<td><strong>Neural Network Theory</strong>: Neuronal Dynamics: Activations and signals, Neurons as functions, signal monotonicity, Biological Activations and signals, Neuron Fields, Neuron Dynamical Systems, Common signal functions, Pulse-Coded Signal functions</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>IV</td>
<td><strong>Neural Networks and Fuzzy systems</strong>: Neural and fuzzy machine Intelligence, Fuzziness as Multivalence, The Dynamical Systems approach to Machine Intelligence, The brain as a dynamical system, Neural and fuzzy systems as function Estimators, Neural Networks as trainable Dynamical system, Fuzzy systems and applications, Intelligent Behaviour as Adaptive Model free Estimation, Generalization and creativity, Learning as change, Symbol vs Numbers, Rules vs Principles, Expert system Knowledge as rule trees, Symbolic vs Numeric Processing, Fuzzy systems as Structured Numerical estimators, Generating Fuzzy rules with product space Clustering, Fuzzy Systems as Parallel associators, Fuzzy systems as Principle based Systems</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Genetic Algorithms</strong>: A simple genetic algorithm, A simulation by hands, similarity templates(Schemata), Mathematical foundations, Schema Processing at work, The two-armed and k-armed Bandit Problem, The building block hypothesis, The minimal Deceptive Problem, Computer implementation of Genetic algorithm, Data Structures, Reproduction, Cross over</td>
<td>4</td>
</tr>
</tbody>
</table>
and Mutation, Time to reproduce and time to Cross Mapping objective function to fitness form, Fitness scaling, Applications of genetic algorithm, De Jong and Function Optimization, Improvement in basic techniques, Introduction to Genetics based machine learning, applications of genetic based machine leaning

### Data Mining:
- Introduction to Data Mining, Computer systems that can learn, Machine learning and methodology of science, Concept learning, Data warehouse, designing decision support systems, Client server and data warehousing, Knowledge Discovery Process, Visualization Techniques, K-nearest neighbor, Decision trees, OLAP tools, Neural networks, Genetic algorithm, Setting up a KDD environment, Real life applications, Customer profiling, Discovering foreign key relationships

### Textbook
1. Introduction to Artificial Intelligence By Eugene Charniak, Drew McDermott- Addison Wesley
2. Neural Networks and fuzzy systems A dynamical systems approach to machine Intelligence by Bart Kosko- PHI
4. Data Mining by Pieter Adriaans and Dolf Zantinge – Pearson Education Asia

Data Warehousing in the Real World by Sam Anahory and Dennis Murray, Addison –Wesley
### Course Code: PSIT303

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Unit</th>
<th>Description</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSIT303</td>
<td>I</td>
<td><strong>Introduction</strong>: Parallel Processing Architectures: Parallelism in sequential machines, Abstract model of parallel computer, Multiprocessor architecture, Pipelining, Array processors. <strong>Programmability Issues</strong>: An overview, Operating system support, Types of operating systems, Parallel programming models, Software tools</td>
<td></td>
</tr>
<tr>
<td></td>
<td>II</td>
<td><strong>Data Dependency Analysis</strong>: Types of dependencies loop and array dependences, Loop dependence analysis, Solving diophantine equations, Program transformations <strong>Shared Memory Programming</strong>: General model of shared memory programming, Process model under UNIX</td>
<td></td>
</tr>
<tr>
<td></td>
<td>III</td>
<td><strong>Algorithms for Parallel Machines</strong>: Speedup, Complexity and cost, Histogram computation, Parallel reduction, Quadrature problem, Matrix multiplication, Parallel sorting algorithms, Solving linear systems, Probabilistic algorithms <strong>Message Passing Programming</strong>: Introduction, Model, Interface, Circuit satisfiability, Introducing collective, Benchmarking parallel performance</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>IV</td>
<td><strong>Parallel Programming languages</strong>: Fortran90, nCUBE C, Occam, C-Linda <strong>Debugging Parallel Programs</strong>: Debugging techniques, Debugging message passing parallel programs, Debugging shared memory parallel programs <strong>Memory and I/O Subsystems</strong>: Hierarchical memory structure, Virtual memory system, Memory allocation and management, Cache allocation and management, Cache memories and management, Input output subsystems</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V</td>
<td><strong>Other Parallelism Paradigms</strong>: Data flow computing, Systolic architectures, Functional and logic paradigms, Distributed shared memory <strong>Performance of Parallel Processors</strong>: Speedup and efficiency, Amdahl’s law, Gustafson-Barsis’s law, Karf-Flatt metric, Isoefficiency metric</td>
<td></td>
</tr>
</tbody>
</table>

### Books:
2. Jorden H. F. and Alaghband G., “*Fundamentals of Parallel Processing*” M.J. Quinn, “*Parallel Programming*”, TMH

### References:
1. Shasikumar M., “*Introduction to Parallel Processing*”, PHI
2. Wilson G.V., “*Practical Parallel Programming*”, PHI
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Unit</th>
<th>Description</th>
<th>Credits</th>
</tr>
</thead>
</table>
| PSIT304     | I    | Defining the scope of multimedia, Hypertext and Collaborative research, Multimedia and personalised computing, Multimedia on the map, Emerging applications, The challenges  

**The convergence of computers, Communications, and entertainment products**  
The technology trends, Multimedia appliances, Hybrid Devices, Designers perspective, industry perspective of the future, Key challenges ahead, Technical, regulatory, Social  

**Architectures and issues for Distributed Multimedia systems**  
Distributed Multimedia systems, Synchronization, and QOS Architecture, The role of Standards, A framework for Multimedia systems                                                                                                                                                                                                                                                                                                                                                     | 4       |
| PSIT304     | II   | **Digital Audio Representation and processing**  
Uses of Audio in Computer Applications, Psychoacoustics, Digital representation of sound, transmission of digital sound, Digital Audio signal processing, Digital music making, Speech recognition and generation, digital audio and the computers  
Video Technology  
Raster Scanning Principles, Sensors for TV Cameras, Colour Fundamentals, Colour Video, Video performance Measurements, Analog video Artifacts, video equipments, World wide television standards                                                                                                                                                                                                                                            | 4       |
| PSIT304     | III  | **Digital Video and Image Compression**  
Video compression techniques, standardization of Algorithm, The JPEG Image Compression Standard, ITU-T Recommendations, The EPEG Motion Video Compression Standard, DVI Technology  
**Operating System Support for Continuous Media Applications**  
Limitation of Work station Operating system, New OS support, Experiments Using Real Time Mach  
**Middleware System Services Architecture**  
Goals of Multimedia System services, Multimedia system services Architecture, Media stream protocol  
**Multimedia Devices, Presentation Services, and the User Interface**  
Client control of continuous multimedia, Device control, Temporal coordination and composition, toolkits, hyperapplications                                                                                                                                                                                                                                                                                                                                                     | 4       |
<table>
<thead>
<tr>
<th>IV</th>
<th>Multimedia File systems and Information Models</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The case for multimedia information systems, The file system support for continuous Media, Data models for multimedia and Hypermedia information, Content-based Retrieval of Unstructured Data</td>
</tr>
<tr>
<td></td>
<td><strong>Multimedia presentation and Authoring</strong></td>
</tr>
<tr>
<td></td>
<td>Design paradigms and User interface, barriers to widespread use, research trends</td>
</tr>
<tr>
<td></td>
<td><strong>Multimedia Services over the Public Networks</strong></td>
</tr>
<tr>
<td></td>
<td>Requirements, Architecture, and protocols, Network services, applications</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>V</th>
<th>Multimedia Interchange</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Quick time Movie File Format, QMFI, MHEG (Multimedia and Hypermedia Information Encoding Expert Group), Format Function and representation, Track model and Object model, Real Time Interchange</td>
</tr>
<tr>
<td></td>
<td><strong>Multimedia conferencing</strong></td>
</tr>
<tr>
<td></td>
<td>Teleconferencing Systems, Requirements of Multimedia Communications, Shared Application Architecture and embedded Distributed objects, Multimedia Conferencing Architecture</td>
</tr>
<tr>
<td></td>
<td><strong>Multimedia Groupware</strong></td>
</tr>
<tr>
<td></td>
<td>Computer and Video fusion approach to open shared workspace, High Definition Television and desktop computing, HDTV standards, Knowledge based Multimedia systems, Anatomy of an Intelligent Multimedia system</td>
</tr>
</tbody>
</table>

**Text Book**

Multimedia Systems by John F. Koegel Buford- Pearson Education
Practical Components:

Practicals:
1. You have got the brilliant idea of setting up a company that sells testing services to software houses. Make a strategic plan for your company, taking into account the following issues:
   - What is the testing process that will be followed in the company?
   - What is the focus of the testing services?
   - What kind of people are you going to hire as staff for the company?
   - How are you going to validate that a testing project carried out in the company has been beneficial to the customer?
   - What kind of automated tools will the company use?
2. Prepare a small project and submit SRS, design, coding and test plan.
3. The program reads an arbitrary number of temperatures (as integer numbers) within the range -60°C … +60°C and prints their mean value. Design test cases for testing the program with the black-box strategy.
4. When getting a person’s weight and height as input, the program prints the person’s body weight index. The weight is given in kilograms (as a real number, for instance: 82.0) and the height in meters (as a real number, for instance: 1.86). The body weight index equals weight divided by height squared: weight / (height ° height). Design test cases for testing the program with the black-box strategy.
5. Let us study the following program:
   ```
   x=0; read(y);
   while (y > 100) { x=x+y; read(y); }
   if (y < 200) print(x) else print(y);
   ```
   a) Construct a control-flow graph for the program.
   b) Design test cases for reaching complete branch coverage over the program. Use as few test cases as possible.
6. Design test cases for the following program with the “simple loop” strategy:
   ```
   x=0; read(y);
   while ((y > 100) && (x < 10)) { x=x+1; read(y); }
   print(y);
   ```
   a) Construct a data-flow graph for the program with respect to variable x.
   b) Which execution paths have to be traversed during testing, in order to reach complete all-definition coverage with respect to variable x?
   c) Minimize the number of paths and tests.
   d) Which execution paths have to be traversed during testing, in order to reach complete all-uses coverage with respect to variable x? Minimize the number of paths and tests.
   e) Design test cases for reaching the (minimal) complete all-uses coverage with respect to variable x.
7. Manual Testing for the project
   a) Walkthrough
   b) Whitebox Testing
   c) Blackbox Testing
   d) Unit Testing
   d) Integration Testing
8. Functional Testing
   a) Boundary value Testing
   b) Equivalence class testing
   c) Decision Table based testing
   d) Cause-effect graphs
10. Automated Testing for websites
    a) Load Testing
    b) Performance Testing
Practicals:
1. Write a program using C/C++/Java for implementing the Depth First Search Algorithm.
2. Write a program using C/C++/Java for implementing the Breadth First Search Algorithm.
3. Write and execute the statements for the following in LISP:
   i. Define the function average, which will take two numbers as arguments and compute their average.
   ii. Define the function that computes the factorial of a number.
4. Write and execute the statements for the following in LISP:
   i. Define the function average, which will take two numbers as arguments and compute their average.
   ii. Define the function that computes the factorial of a number.
5. Write Lisp program to enter a list of integers between two given integers and generate new lists by performing different list manipulation functions like first, first, rest, last, append, reduce, reverse, combs, reverse etc.
6. Write a program in Lisp to enter a number n and create a list of length n of repeated elements. Count the number of repetitions of each element in the list. Display the count value of each element in character.
   (Eg (1 2 1 1 3) should give the output “element 1: three times”).
7. Write a program in Lisp to demonstrate working of an artificial neuron.
   (Enter an input vector X and weight vector W. Calculate weighted sum XW. Transform this using signal or activation functions like logistic, threshold, hyperbolic-tangent, linear, exponential, sigmoid etc. and display the output.)
8. Write a program in Lisp to demonstrate that a fuzzy system is a structured numerical estimator-using example of controlling an inverted pendulum.
   (Enter two strings for angle of pendulum θ and angular velocity Δθ. Convert them into strings of fuzzy using five fuzzy set values NM (Negative medium), NS (Negative Small), ZE(Zero), PS (Positive Small) and PM (Positive Medium). Apply FAM rules to find the output values v, the current to the motor control of the pendulum to each input set of (θ, Δθ). Create a string of fuzzy output values of the current.)
9. Write a program in Lisp to perform roulette wheel selection. (Enter a population of binary strings. Consider fitness value as the number of repetitions of a specified bit (say 1) in the string.)
10. Write a program in Lisp to demonstrate the genetic operator mutation.
    (Enter a population of binary strings. Perform mutation based on a random mutation point. Consider fitness value as the number of repetitions of a specified bit (say 1) in the string. Perform the evolution for a specified number generation.)
11. Write a program in Lisp to demonstrate the crossover genetic operator. (Enter population of binary strings. Perform crossover based on a random crossing site value. Consider fitness value as the number of repetitions of specified bit in the string. Perform the evolution for a specified number generation).
12. Write a genetic algorithm program in Lisp to evolve a word with non-repetitive character (eg ‘computer’) by taking a population size of say 5 and performing mutation and crossover. (Use a suitable fitness function, which acknowledges the correct character at the correct location).
13. Write a program in Lisp to implement decision tree. (Enter a decimal string of repeated entries of suitable length. Apply binary decision tree to find repetitions of each decimal digit in the string.)
<table>
<thead>
<tr>
<th>PSITP11 Practical</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Demonstrate the creation and destruction of processes under *nixes.</td>
<td></td>
</tr>
<tr>
<td>2. Demonstrate the concept of Mutual exclusion in parallel processing</td>
<td></td>
</tr>
<tr>
<td>3. Demonstrate the shared memory model of programming in parallel processing</td>
<td></td>
</tr>
<tr>
<td>4. Solution to Producer Consumer problem using Mutual Exclusion</td>
<td></td>
</tr>
<tr>
<td>5. Creation and killing of processes using custom (shmlib.h) library</td>
<td></td>
</tr>
<tr>
<td>6. Sum of elements of an array using loop splitting</td>
<td></td>
</tr>
<tr>
<td>7. Multiplication of 2x2 matrices using loop splitting</td>
<td></td>
</tr>
<tr>
<td>8. Addition of 3x3 Matrices using loop splitting</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PSITP12 1) Create an audio applet or swing based application with play, pause and stop options.</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>• An applet is created. An applet will show three buttons-play, loop and stop.</td>
<td></td>
</tr>
<tr>
<td>• Whenever user clicks on play button an audio file with .wav extension starts playing.</td>
<td></td>
</tr>
<tr>
<td>• Whenever user clicks on loop button, same audio file will be played again and again.</td>
<td></td>
</tr>
<tr>
<td>• User can click on stop button to stop the audio file.</td>
<td></td>
</tr>
<tr>
<td>• Create a video applet or swing based application which will show the video file (.mpg format)</td>
<td></td>
</tr>
</tbody>
</table>

2) Create an applet or swing based application to play separate streams (video and audio) in synchronism. |
| • An applet is created, which will show the audio file (.wav format) |
| • Another applet is created, which will show the video file (.mpg format) |
| • For this we have to install java media framework (java media player) |
| • Two separate threads are created for video and audio files. |

3) Write a program based on applet or swing based application which will display the list of all the alphabets in a combo. Whenever the user selects one of the alphabets corresponding image with audio should be displayed. For eg. If user selects ‘A’ then the audio “A for Apple” should be played. |

4) Write a program to mix different audio streams. Use separate thread for each audio file (.mp3 files) |

5) Write a program to teach counting the objects (with the help of images and audio). When the user clicks on any of the given buttons (Triangle, Rectangle, Circle or Square) respective shape should be drawn. The corresponding Audio file should be played. The shapes and the corresponding audio file should be played the number of times user has clicked the button. |

6) Write a program to design a calculator with audio for basic operations like addition, subtraction, multiplication and division. When the user clicks on any of the number (0 to 9) or operator (+,-, clear), it should be displayed in the textbox with corresponding audio. Simultaneously the output should be displayed in the textbox along with the corresponding audio. For E.g. 3 +4 =7 audio will be “three plus four equal to seven”. |
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 7) Using Media tracker, create a slide show.  
   • Create different frames.  
   • Make the use of media tracker to create a slide show for the created frames.  
| 8) Provide the buttons start, stop, previous and next, so that the user should be able to start the slideshow, go to the previous slide, go to the next slide and stop the presentation.  
| 9) Using Media tracker, create a movie clip.  
   • Create different frames.  
   • Make the use of media tracker to create a movie clip for the created frames.  
   • The clip should be of at least 4 sec.  |
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Unit</th>
<th>Description</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSIT401</td>
<td>I</td>
<td><strong>Introduction:</strong> Security, Attacks, Computer criminals, Method of defence</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Program Security:</strong> Secure programs, Non-malicious program errors, Viruses and other malicious code, Targeted malicious code, Controls against program threats</td>
<td></td>
</tr>
<tr>
<td></td>
<td>II</td>
<td><strong>Operating System Security:</strong> Protected objects and methods of protection, Memory address protection, Control of access to general objects, File protection mechanism, Authentication: Authentication basics, Password, Challenge-response, and Biometrics.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>III</td>
<td><strong>Database Security:</strong> Security requirements, Reliability and integrity, Sensitive data, Interface, Multilevel database, Proposals for multilevel security</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>IV</td>
<td><strong>Security in Networks:</strong> Threats in networks, Network security control, Firewalls, Intrusion detection systems, Secure e-mail, Networks and cryptography, Example protocols: PEM, SSL, IPsec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V</td>
<td><strong>Administrating Security:</strong> Security planning, Risk analysis, Organizational security policies, Physical security. <strong>Legal, Privacy, and Ethical Issues in Computer Security:</strong> Protecting programs and data, Information and law, Rights of employees and employers, Software failures, Computer crime, Privacy, Ethical issues in computer society, Case studies of ethics</td>
<td></td>
</tr>
<tr>
<td>Course Code</td>
<td>Unit</td>
<td>Description</td>
<td>Credits</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>Direct Kinematics: Dot and cross products, Co-ordinate frames, Rotations, Homogeneous, Co-ordinates, Link co-ordination arm equation, (Five-axis robot, Four axis robot, Six axis robot).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>III</td>
<td>Inverse Kinematics: General properties of solutions tool configuration Five axis robots, Three-Four axis, Six axis robot (Inverse kinematics). Workspace analysis and trajectory planning work envelop and examples, workspace fixtures, Pick and place operations, Continuous path motion, Interpolated motion, Straight-line motion.</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>IV</td>
<td>Robot Vision: Image representation, Template matching, Polyhedral objects, Shane analysis, Segmentation (Thresholding, region labelling, Shrink operators, Swell operators, Euler numbers, Perspective transformation, Structured Illumination, Camera calibration).</td>
<td></td>
</tr>
<tr>
<td>Course Code</td>
<td>Unit</td>
<td>Description</td>
<td>Credits</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>PSIT403</td>
<td>I</td>
<td><strong>Introduction to Distributed System:</strong> Goals, Hardware concepts, Software concepts, and Client-Server model. Examples of distributed systems.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Communication:</strong> Layered protocols, Remote procedures call, Remote object invocation, Message-oriented communication, Stream-oriented communication.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>II</td>
<td><strong>Processes:</strong> Threads, Clients, Servers, Code Migration, Software agent. <strong>Naming:</strong> Naming entities, Locating mobile entities, Removing un-referenced entities.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>III</td>
<td><strong>Synchronization:</strong> Clock synchronization, Logical clocks, Global state, Election algorithms, Mutual exclusion, Distributed transactions. <strong>Consistency and Replication:</strong> Introduction, Data centric consistency models, Client centric consistency models, Distribution protocols, Consistency protocols.</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>IV</td>
<td><strong>Fault Tolerance:</strong> Introduction, Process resilience, Reliable client server communication, Reliable group communication. Distributed commit, Recovery. <strong>Security:</strong> Introduction, Secure channels, Access control, Security management.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V</td>
<td><strong>Distributed File System:</strong> Sun network file system, CODA files system. <strong>Case Study:</strong> CORBA, Distributed COM, Globe, Comparison of CORBA, DCOM, and Globe.</td>
<td></td>
</tr>
<tr>
<td>Course Code</td>
<td>Unit</td>
<td>Description</td>
<td>Credits</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| PSIT404     | I    | **Java Programming**  
Object oriented programming revisited, JDK, Java Virtual machine-Platform independent-portability-scalability  
Operators and expressions-decision making, branching, looping, Classes, Objects and methods, Arrays Strings and Vectors, Interfaces, Packages, Multi-Threading, managing errors and exceptions, Applet programming, Managing files and streams |         |
|             | II   | **Java Technology for Active Web Documents**  
An Early Form of Continuous Update, Active Documents and Server Overhead, Active Document Representation and Translation, Java Technology, the Java Run-Time Environment, The Java Library  
A Graphics Toolkit, Using Java Graphics on a Particular Computer, Java Interpreters and Browsers  
Compiling a Java Program, Invoking an Applet, Example of Interaction with a Browser |         |
|             | III  | **RPC and Middleware**  
Programming Clients and Servers, Remote Procedure Call Paradigm, RPC Paradigm, Communication Stubs, External Data Representation, Middleware and Object-Oriented Middleware | 4       |
|             | IV   | **Network Management (SNMP)**  
Managing an Internet, The Danger of Hidden Features, Network Management Software, Clients, Servers, Managers and Agents, Simple Network Management Protocol, Fetch-Store Paradigm, The MIP and Object Names, The Variety of MIB Variables, MIB variables that correspond to arrays |         |
|             | V    | **Java technologies**  
Graphics, JFC-JAVA foundation classes, swing, images, java 2d graphics, internationalization, Communication and Networking, TCP Sockets, UDP Sockets, java.net, java security, Object serialization, Remote method serialization, JDBC: Java Data Base Connectivity, Java beans, Java interface to CORBA, JAVA- COM Integration, Java Media Framework, commerce and java wallet, Data structures and java utilities, JavaScript, Servelets |         |
<table>
<thead>
<tr>
<th>PSITP13</th>
<th>Project</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSITP14</td>
<td>Project</td>
<td>2</td>
</tr>
</tbody>
</table>
| PSITP15 | 1. Demonstrate Network Operating System by implementing stack  
2. Demonstrate Network Operating System by implementing queue  
3. Demonstrate Network Operating System by implementing stack/queue for sorting numbers  
4. Calculator using Remote Method Invocation  
5. Time and Date from server, using TCP  
6. *Demonstrate Java Native Interface*  
7. Compute values of equations using Remote Method Invocation  
8. *Chat server / client system using TCP*  
9. Demonstrate Distributed COM  
Accessing Remote Database using Remote Method Invocation | 2 |
| PSITP16 | 1. Design a program to implement a concept of class and constructor and inheritance  
   Design a class Bank to represent the bank account having following data members  
   1. account number  
   2. password  
   3. account type  
   4. customer name  
   5. balance amount  
   And member functions are  
   1. getWithdraw()  
   2. Moneydeposit()  
   3. balanceenquire()  
   Create class XXXbank which inherits the properties of above class and creates the number of customer as per the XXXBank requirement (max. customer the bank can hold) and allow the customer to use account.  
   **Note:** password of customer should be invisible(Using Console class)  
2. Java code to display MIB node information using com.adventnet.snmp.mibs package  
3. Applet application  
   a. write a applet application which displays the name in a serif font in plain bold and bold italic. Name should be displayed on the top of each other.  
   b. Write a program to draw a different shape using graphics 2D and Shape interface as per the user co-ordinates. And also paint then using colors and give an effect of transparency, cyclic coloring and different color pattern also mention the line stroke of the shape as doted etc.  
4. Write a program for multithreading to bounce a balls of different randomly changing colors. For each ball separate thread is assign.  
5. Program for file handling to implant the ceaser cipher an | 2 |
<table>
<thead>
<tr>
<th>No.</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.</td>
<td>Design a user define bean name as <strong>Thermometer</strong>. Which is use to check the temperature. Also define the <strong>TemperatureListener</strong> and <strong>TemperatureEvent</strong> class for it. Temperature Listener contain one handler called <strong>temperatureChanged</strong> which is called when thermometer temperature changes.</td>
</tr>
<tr>
<td>7.</td>
<td>Swing application to design a editor like word where we can type a text or draw picture and we can save it.</td>
</tr>
</tbody>
</table>
| 8.  | JDBC:  
   a. Write a java code to create a table named student and insert the records in it.  
   b. Also alter the table by adding new column named photo where you can add student photo.  
   c. Also to retrieve the records from the data base. |
| 9.  | Java Localization (Using Local class, ResourceBundle) to get a concept of localizing program as per the locals.  
   a. Displaying entered name in different languages.  
   b. Displaying currency or number in different locals. |
| 10. | JSP:  
   a. Write a JSP program to display a bean.  
   b. Write a JSP code to validate an email address. |