Bachelor of Technology (Computer Science and Engineering)

Semester-VI

L-0 T-0 P-2 C-4

CSE330PR1: Software Engineering

Course Objectives

1. To provide the idea of decomposing the given problem into Analysis, Design, Implementation, Testing and Maintenance phases.
2. To provide an idea of using various process models in the software industry according to given circumstances.
3. To gain the knowledge of how Analysis, Design, Implementation, Testing and Maintenance processes are conducted in a software project.

Course Outcomes (COs)

1. Students will be able to decompose the given project in various phases of a lifecycle.
2. Students will be able to choose appropriate process model depending on the user requirements.
3. Students will be able perform various life cycle activities like Analysis, Design, Implementation, Testing and Maintenance.
4. Students will be able to know various processes used in all the phases of the product.
5. Students can apply the knowledge, techniques, and skills in the development of a software product.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	-	-	2	-	-	-	-	-	-	-	-	-	1	-	-
CO2	2	2	-	-	-	-	-	-	-	-	-	-	1	2	-
CO3	2	-	1	-	3	-	-	1	-	-	-	-	2	-	-
CO4	2	2	-	2	-	1	-	1	-	-	-	3	-	-	-
CO5	1	2	-	-	-	-	-	-	-	-	-	3	1	-	-

High-3 Medium-2 Low-1

Practical's:

Lab work should include a running case study problem for which different deliverables at the end of each phase of a software development life cycle are to be developed. This will include modeling the requirements, architecture and detailed design. Subsequently the design models will be coded and tested.

Reference(s)

1. Ian Sommerville, “Software Engineering”, 9th Edition, Pearson Education Asia, 2011.
2. Rajib Mall, “Fundamentals of Software Engineering”, Third Edition, PHI Learning Private Limited ,2009.
3. Pankaj Jalote, “Software Engineering, A Precise Approach”, Wiley India, 2010.
4. Kelkar S.A., “Software Engineering”, Prentice Hall of India Pvt Ltd, 2007.
5. Stephen R. Schach, “Software Engineering”, Tata McGraw-Hill Publishing Company Limited, 2007.

List of e-Learning Resources:

Software Engineering (NPTEL) by Prof. Rushikesh K Joshi, Prof. Umesh Bellur, Prof. N.L. Sarda (IIT, Bombay)
Link : https://nptel.ac.in/courses/106101061

Bachelor of Technology (Computer Science and Engineering)

Semester-VI

L-0 T-0 P-2 C-1

CSE 812 PR1: Image processing

Course Objectives

- To learn foundational aspects of digital images, the course comprehensively covers resolution, color models, and pixel representation, fostering a profound understanding of image basics
- To empower students in manipulating images for diverse applications, the course extensively explores techniques such as image enhancement, filtering, segmentation, and feature extraction
- To provide practical expertise, students engage in hands-on projects within domains like medical imaging and computer vision, employing acquired concepts to solve practical challenges
- To underscore responsible practices, the course addresses ethical concerns encompassing image manipulation, privacy, and dataset biases in image processing applications
- To anticipate the future of this dynamic field, the course explores emerging trends in deep learning and advanced image processing, encouraging ongoing exploration of cutting-edge technologies

Course Outcomes (COs)

1. Understanding the digital image fundamentals, encompassing image representation, resolution, and color models
2. Apply diverse image processing techniques like enhancement, filtering, segmentation, and feature extraction to effectively manipulate and enhance image quality
3. Create practical skills through hands-on projects, enabling the resolution of real-world image processing challenges across varied domains
4. Understand heightened ethical awareness in image manipulation, addressing privacy concerns and biases in datasets responsibly
5. Analyze emerging trends in deep learning, advanced image processing methodologies, and future technologies, fostering curiosity and encouraging ongoing exploration beyond the course

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	-	-	1	-	-	-	-	-	2	-	2	1
CO2	2	3	1	-	-	-	-	1	-	1	-	1	3	1	3
CO3	1	2	3	1	1	-	-	-	-	1	-	1	3	1	1
CO4	3	-	-	3		-	-	1	-	-	-	1	3	1	2
CO5	3	3		1	2		1	-	-	1	-	2	3	1	1

High-3 Medium-2 Low-1

Practical(s)

1. Experiment with various image resolutions and observe the impact on image quality and file size.
2. Convert images between different color models (RGB, CMYK, HSL, HSV) and analyze the visual differences.
3. Apply Fourier Transform to images and visualize the frequency components in the frequency domain.
4. Perform basic mathematical operations (addition, subtraction, multiplication) on images and observe the effects.
5. Apply histogram equalization to images with varying histograms and evaluate the enhancement impact.
6. Implement contrast stretching algorithms and observe changes in image contrast for different input images.
7. Compare the effectiveness of different noise reduction filters (e.g., median, Gaussian) on noisy images.
8. Apply sharpening and blurring filters to images to understand their effects on image details and smoothness.
9. Implement convolution operations for image filtering with various kernels (e.g., Gaussian, Sobel).

10. Implement different edge detection algorithms (Canny, Sobel, Roberts) and compare their performance on various images.
11. Perform blob analysis to detect and characterize objects within images of varying complexities.
12. Apply texture analysis techniques to classify images based on different textures present.
13. Experiment with global and local thresholding techniques for segmenting images and compare the results.
14. Train a simple machine learning model (e.g., SVM) for image classification and object recognition tasks.
15. Implement a basic CNN for image classification and evaluate its performance on a dataset.
16. Use the Watershed algorithm for region-based segmentation on different types of images.
17. Apply translation, rotation, and scaling operations to images and observe the geometric changes.
18. Implement deblurring and denoising techniques to restore degraded images and assess the restoration quality.
19. Explore and discuss case studies on ethical issues related to image manipulation and forgery.
20. Implement image-to-image translation models (e.g., CycleGAN) and observe the transformation results between different image domains.

Total: 30 Hours

Reference(s)

1. "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods
2. "Image Processing, Analysis, and Machine Vision" by Milan Sonka, Vaclav Hlavac, and Roger Boyle
3. "Computer Vision: Algorithms and Applications" by Richard Szeliski
4. "Understanding Digital Signal Processing" by Richard G. Lyons
5. "Pattern Recognition and Machine Learning" by Christopher M. Bishop
6. "Introduction to Probability and Statistics for Engineers and Scientists" by Sheldon M. Ross
7. "Ethics in Information Technology" by George Reynolds
8. "Computer Vision: A Modern Approach" by David A. Forsyth and Jean Ponce
9. "Digital Image Processing using MATLAB" by Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins
10. "Handbook of Medical Image Processing and Analysis" by Isaac Bankman

List of e-Learning Resources:

1.https://www.geeksforgeeks.org/digital-image-processing-basics/

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Bachelor of Technology (Computer Science and Engineering)

Semester-VI

L-2 T-1 P-0 C-3

CSE 812 TR1: Image processing

Course Objectives

- To learn foundational aspects of digital images, the course comprehensively covers resolution, color models, and pixel representation, fostering a profound understanding of image basics
- To empower students in manipulating images for diverse applications, the course extensively explores techniques such as image enhancement, filtering, segmentation, and feature extraction
- To provide practical expertise, students engage in hands-on projects within domains like medical imaging and computer vision, employing acquired concepts to solve practical challenges
- To underscore responsible practices, the course addresses ethical concerns encompassing image manipulation, privacy, and dataset biases in image processing applications
- To anticipate the future of this dynamic field, the course explores emerging trends in deep learning and advanced image processing, encouraging ongoing exploration of cutting-edge technologies

Course Outcomes (COs)

6. Understanding the digital image fundamentals, encompassing image representation, resolution, and color models
7. Apply diverse image processing techniques like enhancement, filtering, segmentation, and feature extraction to effectively manipulate and enhance image quality
8. Create practical skills through hands-on projects, enabling the resolution of real-world image processing challenges across varied domains
9. Understand heightened ethical awareness in image manipulation, addressing privacy concerns and biases in datasets responsibly
10. Analyze emerging trends in deep learning, advanced image processing methodologies, and future technologies, fostering curiosity and encouraging ongoing exploration beyond the course

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	-	-	1	-	-	-	-	-	2	-	2	1
CO2	2	3	1	-	-	-	-	1	-	1	-	1	3	1	3
CO3	1	2	3	1	1	-	-	-	-	1	-	1	3	1	1
CO4	3	-	-	3		-	-	1	-	-	-	1	3	1	2
CO5	3	3		1	2		1	-	-	1	-	2	3	1	1

High-3 Medium-2 Low-1

UNIT I: Foundations of Digital Images — 10 Hours

Overview of Digital Images: Definition and characteristics of digital images, Historical development of digital images, Applications of digital images in various fields. Image Representation: Resolution: Spatial and Temporal Resolution, Color Models: RGB, CMYK, HSL, HSV, Pixel Representation: Binary, Grayscale, and Color, Image Basics: Spatial Domain: Pixel values, intensity, spatial resolution, Frequency Domain: Fourier transform, frequency components, Mathematical Preliminaries: Matrices, vectors, basic algebra for image processing

UNIT II: Image Enhancement Techniques — 8 Hours

Histogram Equalization: Concept and purpose, Algorithm and implementation, Applications and limitations, Contrast Stretching : Importance of contrast in images, Linear and non-linear contrast stretching techniques, Adaptive contrast stretching, Noise Reduction: Types of noise in images,Filtering techniques: mean filter, median filter, Image smoothing for noise reduction Image Sharpening and Blurring Techniques : Convolution for image sharpening, High-pass and low-pass filters for blurring, Edge enhancement techniques

UNIT III: Image filtering & feature extraction **10 Hours**

Convolution and its applications: Convolution operation in image processing, Applications in image filtering and feature extraction Various image filters: Gaussian filter for smoothing, Sobel and Laplacian filters for edge detection, Importance of kernel size in filtering Edge Detection & boundary detection: Edge detection algorithms: Sobel, Prewitt, Canny, Boundary extraction techniques, Applications of edge detection in image processing Feature extraction techniques: Blob Analysis: Detecting and analyzing connected regions, Texture Analysis: Haralick's texture features, Shape analysis for feature extraction

UNIT IV: Image segmentation and recognition **9 Hours**

Image segmentation techniques:Thresholding techniques, Region-based segmentation, Clustering-based segmentation Object recognition & classification: Object recognition algorithms: template matching, neural networks, Classification techniques: SVM, k-NN, Challenges in object recognition, Introduction to machine learning in Image processing: Supervised and unsupervised learning, Training and testing datasets, Applications of machine learning in image processing Neural networks for image analysis: Basics of neural networks, Convolutional Neural Networks (CNNs), Transfer learning in image analysis

UNIT V: Advanced topics and applications **8 Hours**

Geometric transformation: Translation, rotation, scaling operations, Homogeneous coordinates and transformation matrices Image restoration and reconstruction: Image deblurring techniques, Inverse problems in image restoration, Super-resolution techniques Ethical considerations in image processing: Privacy concerns, Bias and fairness in image analysis, Responsible use of image processing technologies, Emerging trends: Deep Learning Architectures: *Generative Adversarial Networks (GANs), Recurrent Neural Networks (RNNs),* Image-to-Image Translation, Applications and future prospects in image processing.

Total: 45 Hours

Reference(s)

11. "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods
12. "Image Processing, Analysis, and Machine Vision" by Milan Sonka, Vaclav Hlavac, and Roger Boyle
13. "Computer Vision: Algorithms and Applications" by Richard Szeliski
14. "Understanding Digital Signal Processing" by Richard G. Lyons
15. "Pattern Recognition and Machine Learning" by Christopher M. Bishop
16. "Introduction to Probability and Statistics for Engineers and Scientists" by Sheldon M. Ross
17. "Ethics in Information Technology" by George Reynolds
18. "Computer Vision: A Modern Approach" by David A. Forsyth and Jean Ponce
19. "Digital Image Processing using MATLAB" by Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins
20. "Handbook of Medical Image Processing and Analysis" by Isaac Bankman

List of e-Learning Resources:

1.https://www.geeksforgeeks.org/digital-image-processing-basics/

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Bachelor of Technology (Computer Science and Engineering)

Semester-VI

L-2 T-1 P-0 C-4

CSE330TR1: Software Engineering

Course Objectives

1. To provide the idea of decomposing the given problem into Analysis, Design, Implementation, Testing and Maintenance phases.
2. To provide an idea of using various process models in the software industry according to given circumstances.
3. To gain the knowledge of how Analysis, Design, Implementation, Testing and Maintenance processes are conducted in a software project.

Course Outcomes (COs)

6. Students will be able to decompose the given project in various phases of a lifecycle.
7. Students will be able to choose appropriate process model depending on the user requirements.
8. Students will be able perform various life cycle activities like Analysis, Design, Implementation, Testing and Maintenance.
9. Students will be able to know various processes used in all the phases of the product.
10. Students can apply the knowledge, techniques, and skills in the development of a software product.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	-	-	2	-	-	-	-	-	-	-	-	-	1	-	-
CO2	2	2	-	-	-	-	-	-	-	-	-	-	1	2	-
CO3	2	-	1	-	3	-	-	1	-	-	-	-	2	-	-
CO4	2	2	-	2	-	1	-	1	-	-	-	3	-	-	-
CO5	1	2	-	-	-	-	-	-	-	-	-	3	1	-	-

High-3 Medium-2 Low-1

UNIT I: Software Process and Project Management **7 Hours**

Software process and project management: Introduction to Software Engineering, Software Process, Perspective and Specialized Process Models – Software Project Management: Estimation – LOC and FP Based Estimation, COCOMO Model – Project Scheduling – Scheduling, Earned Value Analysis - Risk Management.

UNIT II: Requirements Analysis and Specification **11 Hours**

Requirements analysis and specification : Software Requirements: Functional and Non-Functional, User requirements, System requirements, Software Requirements Document – Requirement Engineering Process: Feasibility Studies, Requirements elicitation and analysis, requirements validation, requirements management-Classical analysis: Structured system Analysis, Petri Nets- Data Dictionary.

UNIT III: Software Design **7 Hours**

Design process – Design Concepts-Design Model– Design Heuristic – Architectural Design – Architectural styles, Architectural Design, Architectural Mapping using Data Flow- User Interface Design: Interface analysis, Interface Design –Component level Design: Designing Class based components, traditional Components.

UNIT IV: Testing and Implementation **11 Hours**

Testing and implementation: Software testing fundamentals -Internal and external views of Testing-white box testing- basis path testing-control structure testing-black box testing- Regression Testing – Unit Testing – Integration Testing – Validation Testing – System Testing And Debugging – Software Implementation Techniques: Coding practices-Refactoring.

UNIT V: Project Management **9 Hours**

Project management Estimation – FP Based, LOC Based, Make/Buy Decision, COCOMO II - Planning – Project Plan, Planning Process, RFP Risk Management – Identification, Projection, RMMM

- Scheduling and Tracking –Relationship between people and effort, Task Set & Network, Scheduling, EVA – Process and Project Metrics.

Practical's:

Lab work should include a running case study problem for which different deliverables at the end of each phase of a software development life cycle are to be developed. This will include modeling the requirements, architecture and detailed design. Subsequently the design models will be coded and tested.

Total: 45 Hours

Reference(s)

1. Ian Sommerville, “Software Engineering”, 9th Edition, Pearson Education Asia, 2011.
2. Rajib Mall, “Fundamentals of Software Engineering”, Third Edition, PHI Learning Private Limited ,2009.
3. Pankaj Jalote, “Software Engineering, A Precise Approach”, Wiley India, 2010.
4. Kelkar S.A., “Software Engineering”, Prentice Hall of India Pvt Ltd, 2007.
5. Stephen R. Schach, “Software Engineering”, Tata McGraw-Hill Publishing Company Limited, 2007.

List of e-Learning Resources:

Software Engineering (NPTEL) by Prof. Rushikesh K Joshi, Prof. Umesh Bellur, Prof. N.L. Sarda (IIT, Bombay)
Link : https://nptel.ac.in/courses/106101061

Bachelor of Technology (Computer Science and Engineering)

Semester-VI

L-3 T-1 P-0 C-4

CSE900 TR1: Network and Cyber Security

Course Objectives

- To learn basics of Cryptography and Network Security.
- To learn to develop cyber security strategies and policies.
- To be able to secure a message over insecure channel by various means.
- To learn technical skills to protect and defend against cyber threats.
- To learn about how to maintain the Confidentiality, Integrity and Availability of a data.
- To learn various protocols for network security to protect against the threats in the networks

Course Outcomes (COs)

4. Analyze and evaluate the cyber security needs of an organization.
5. Apply learn about the cyber security threat landscape and the types of cyber attacks, vulnerabilities, and remedies.
6. Analyze software vulnerabilities and security solutions to reduce the risk of exploitation.
7. Evaluate the performance and troubleshoot cyber security systems.
8. Apply cyber security solutions and use of cyber security, information assurance, and cyber/computer forensics software/tools.
9. Apply risk treatment methods and monitor key risk and performance indicators.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	-	1	-	2	-	2	1	1	-	2	3	3	2
CO2	3	2	-	-	1	2	-	2	-	-	1	1	2	-	-
CO3	3	2	2	2	-	3	2	-	2	-	3	-	3	3	2
CO4	3	2	2	3	2	1	-	2	-	2	2	1	3	2	2
CO5	3	3	-	1	2	-	1	-	-	1	2	2	3	1	1

High-3 Medium-2 Low-1

UNIT I: (Introduction to Cryptography and Block Ciphers) 9 Hours

Introduction to security attacks - services and mechanism - introduction to cryptography -Conventional Encryption: Conventional encryption model - classical encryption techniques -substitution ciphers and transposition ciphers – cryptanalysis – steganography - stream and block ciphers - Modern Block Ciphers: Block ciphers principals - Shannon's theory of confusion and diffusion - fiestal structure - data encryption standard(DES) - strength of DES – differential and linear crypt analysis of DES - block cipher modes of operations - triple DES – AES.

Unit II (Confidentiality and Modular Arithmetic) 9 Hours

Confidentiality using conventional encryption - traffic confidentiality - key distribution – random number generation - Introduction to graph - ring and field - prime and relative prime numbers - modular arithmetic - Fermat's and Euler's theorem - primality testing - Euclid's Algorithm - Chinese Remainder theorem - discrete algorithms.

Unit III (Public key cryptography and Authentication requirements) 9 Hours

Principles of public key crypto systems - RSA algorithm - security of RSA - key management – Diffle-Hellman key exchange algorithm - introductory idea of Elliptic curve cryptography – Elgamel encryption - Message Authentication and Hash Function: Authentication requirements - authentication functions - message authentication code - hash functions - birthday attacks – security of hash functions and MACS.

Unit IV (Integrity checks and Authentication algorithms) **9 Hours**

MD5 message digest algorithm - Secure hash algorithm (SHA) Digital Signatures: Digital Signatures - authentication protocols - digital signature standards (DSS) - proof of digital signature algorithm - Authentication Applications: Kerberos and X.509 - directory authentication service - electronic mail security-pretty good privacy (PGP) - S/MIME.

Unit V (IP Security and Key Management) **9 Hours**

IP Security: Architecture - Authentication header - Encapsulating security payloads – combining security associations - key management. Web Security: Secure socket layer and transport layer security - secure electronic transaction (SET) - System Security: Intruders - Viruses and related threads - firewall design principals –trusted systems.

Total: 45 Hours

Reference(s)

1. W. Mao, “Modern Cryptography – Theory and Practice”, Pearson Education.
2. Charles P. Pfleeger, Shari Lawrence Pfleeger – Security in computing – Prentice Hall of India.
3. William Stallings, “Crpyptography and Network security Principles and Practices”, Pearson/PHI.
4. Wade Trappe, Lawrence C Washington, “ Introduction to Cryptography with coding theory”, Pearson.

List of e-Learning Resources:

1. http://nptel.ac.in/courses/106105031/lecture by Dr. Debdeep MukhopadhyayIIT Kharagpur
2. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-033- computer-system-Engineering-spring-2009/video-lectures/ lecture by Prof. Robert Morris and Prof. Samuel Madden MIT.

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Bachelor of Technology (Computer Science and Engineering)
Semester-VI BCT

L-2 T-1 P-0 C-3

CSE910 TR1: Mobile Application Development with Android

Course Objectives

- To facilitate students to understand android SDK
- To help students to gain a basic understanding of Android application development
- To inculcate working knowledge of Android Studio development tool
- Creating robust mobile applications and learn how to integrate them with other services

Course Outcomes (COs)

10. Understand the mobile technology with android
11. Build an application using Android development environment.
12. Experiment with the method of storing, sharing and retrieving the data in Android Applications.
13. Examine responsive user interface across wide range of devices.
14. Create a mobile Application by using various components like activity, views, services, content providers and receivers.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	3	-	2	1	-	2	-	2	1	-	-	2	-
CO2	1	3	-	-	2	-	1	2	-	2	-	-	1	2	-
CO3	1	3	1	1	2	-	3	-	2	-	2	-	3	-	2
CO4	1	-	2	3	-	2	2	1	-	2	-	2	2	1	-
CO5	1	2	-	2	1	-	2	-	2	1	-	2	1	2	2

High-3 Medium-2 Low-1

UNIT I: Introduction to android operating systems **9 Hours**

History of android, introduction to android operating systems, android development tools, Android architectural overview, standard development environment for android applications, anatomy of an android application, creating a new android application, the android project structure. Frameworks and Tools, Programming in the mobile environment, Development Tools: Installing and using Eclipse with ADT plug-in.

Unit II: Emulator – android **9 Hours**

Emulator - android Emulator - android virtual device, launching emulator, editing emulator settings. Installing Virtual machine for Android, configuring the installed tools, dalvik virtual machine & .apk file extension, fundamentals: basic building blocks, activities, services, broadcast receivers & content providers, Application structure, androidmanifest.xml, uses-permission & uses-sdk, resources & R.java, sdk layouts & draw able resources, activities and activity lifecycle, first sample application.

Unit III: Basic UI design **9 Hours**

Basic UI design : form widgets, text fields, layouts, shared preferences preferences from xml, menu : option menu, context menu, sub menu, menu from xml, menu via code, explicit intents, implicit intents, UI design : time and date, images and media, composite, alert, dialogs & toast popup, tabs and tab activity, styles & themes, applying themes via code and manifest file

Unit IV: Event driven programming in android **9 Hours**

Event driven programming in android (text edit, button clicked), SQLite programming, sqlite open helper, sqlite databse, cursor, reading and updating contacts, reading bookmarks, Android debug bridge (adb) tool,

adapters: - arrayadapters, baseadapters, listview and listactivity, custom listview, gridview using adapters, gallery using adapters

Unit V Graphics **9 Hours**

Graphics: Performance and Multithreading, Graphics and UI Performance, Android Graphics and Multimedia, Performance, Memory Management, Android Notifications and Alarms. Audio playback and Media Player, SoundPool, Mobile Agents and Peer-to-Peer Architecture, using System Services and Web Services, Mobility and Location Based Services. Using Location based services Telephony and SMS services Bluetooth, network and Wi-Fi, Camera, Accessing Internet and web services from android applications.

Reference(s)

1. Joseph Annuzzi Jr, Lauren Darcey, Shane Condor, "Advanced Android Application Development, Developers Library", Pearson Education, 4th Edition (2015)
2. Lauren Darcey, Shane Condor, "Android, Wireless Application Development", Pearson Education, 3rd Edition.
3. Paul Deitel, Harvey Deitel, Alexander Wald, "Android 6 for programmers, An AppDriven Approach", Pearson Education
4. Rap Payne, "Beginning App Development with Flutter: Create Cross-Platform Mobile Apps", Apress (2019)

List of e-Learning Resources:

1. https://www.geeksforgeeks.org/introduction-to-android-development/

Subject Tr. Academic Coordinator HoD Sr. Faculty Nominated by DOAA

Bachelor of Technology (Computer Science and Engineering)
Semester-VI BCT

L-2 T-1 P-0 C-3

CSE910 PR1: Mobile Application Development with Android

Course Objectives

- To facilitate students to understand android SDK
- To help students to gain a basic understanding of Android application development
- To inculcate working knowledge of Android Studio development tool
- Creating robust mobile applications and learn how to integrate them with other services

Course Outcomes (COs)

15. Understand the mobile technology with android
16. Build an application using Android development environment.
17. Experiment with the method of storing, sharing and retrieving the data in Android Applications.
18. Examine responsive user interface across wide range of devices.
19. Create a mobile Application by using various components like activity, views, services, content providers and receivers.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	3	-	2	1	-	2	-	2	1	-	-	2	-
CO2	1	3	-	-	2	-	1	2	-	2	-	-	1	2	-
CO3	1	3	1	1	2	-	3	-	2	-	2	-	3	-	2
CO4	1	-	2	3	-	2	2	1	-	2	-	2	2	1	-
CO5	1	2	-	2	1	-	2	-	2	1	-	2	1	2	2

High-3 Medium-2 Low-1

Practical(s)

1. Installation and setup of Java development kit(JDK), setup android SDK, setup eclipse IDE, setup android development tools (ADT) plugins, create android virtual device.
2. Create “Hello World” application. That will display “Hello World” in the middle of the screen using Text View Widget in the red color.
3. Develop an application for demonstration of android activity life cycle.
4. Develop a Registration page to demonstration of Basic widgets available in android.
5. Develop a sample application with login module. (Check username and password) On successful login, Change TextView “Login Successful”. And on failing login, alert user using Toast “Login fail”.
6. Develop a login application where you will have to validate username and passwords till the username and password is not validated, login button should remain disabled.
7. Develop a Login application as above. Validate login data and display Error to user using setError() method.
8. Implement an application that creates an alert upon receiving a message.
9. Develop an application for demonstration of Relative and Table Layout in android.
10. Develop an application for demonstration of Explicitly Starting New Activity using Intent.
11. Develop an application that will Demonstrate Button onClick() Event and change the TextView Color based on button Clicked.
12. Develop an UI such that, one screen has list of all the types of cars. On selecting of any car name, next screen should show Car details like: name, launched date, company name.
13. Develop a native application that uses GPS location information

Total: 30 Hours

Reference(s)

1. Joseph Annuzzi Jr, Lauren Darcey, Shane Condor, “Advanced Android Application Development, Developers Library”, Pearson Education, 4th Edition (2015)
2. Lauren Darcey, Shane Condor, “Android, Wireless Application Development”, Pearson Education, 3rd Edition.
3. Paul Deitel, Harvey Deitel, Alexander Wald, “Android 6 for programmers, An AppDriven Approach”, Pearson Education
4. Rap Payne, “Beginning App Development with Flutter: Create Cross-Platform Mobile Apps”, Apress (2019)

List of e-Learning Resources:

1. https://www.geeksforgeeks.org/introduction-to-android-development/

Subject Tr. Academic Coordinator HoD Sr. Faculty Nominated by DOAA

Mandsaur University

Bachelor of Technology (Computer Science and Engineering)

Semester-VI

L-2 T-1 P-0 C-3

CSE1051 TR1: Natural Language Processing

Course Objectives

- To learn fundamental tasks in Natural Language Processing (NLP) including syntax, semantics, and pragmatics
- To Gain insights into probability basics and information theory relevant to NLP.
- To explore syntactic analysis techniques including Context-Free Grammars and Dependency Grammar
- Explore Semantic Analysis and Meaning Representation.
- Utilize Lexical Resources and Perform Discourse Analysis.

Course Outcomes (COs): Upon completion of this unit students will be able to:

1. Understanding Natural Language Processing (NLP) tasks in syntax, semantics, and pragmatics, Along with their applications and the role of machine learning techniques
2. Analyze including N-grams, part-of-speech tagging using various techniques
3. Analyze probabilistic parsing techniques methods, context-free grammars, dependency grammar.
4. Analyze NLP techniques like word-sense disambiguation, semantic role labeling, and discourse using both supervised and unsupervised approaches
5. Analyze methodologies including Discourse segmentation, Reference Phenomena, and Coreference Resolution, utilizing lexical resources like WordNet, PropBank, and FrameNet

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3	2	1	-	-	1	-	2	-	-	-	2	-	2	1
CO2	2	3	1	-	-	-	-	1	-	1	-	1	3	1	2
CO3	1	2	3	1	1	-	-	-	-	1	-	1	2	1	1
CO4	3	-	-	3		-	-	1	-	-	-	1	1	1	1
CO5	2	3		1	2		1	-	-	1	-	2	3	1	1

High-3 Medium-2 Low-1

UNIT I: Natural Language Processing — 9 Hours

Natural Language Processing tasks in syntax, semantics, and pragmatics – Issues - Applications - The role of machine learning - Probability Basics –Information theory – Collocations -N-gram Language Models - Estimating parameters and smoothing - Evaluating language models.

UNIT II: Word Level Analysis — 9 Hours

Word Level Analysis: Unsmoothed N-grams, Evaluating N-grams, Smoothing, Interpolation and Backoff – Word Classes, Part-of-Speech Tagging, Rule-based, Stochastic and Transformation based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models.

UNIT III: Syntactic Analysis — 9 Hours

Syntactic Analysis: Context-Free Grammars, Grammar rules for English, Treebanks, Normal Forms for grammar – Dependency Grammar – Syntactic Parsing, Ambiguity, Dynamic Programming parsing – Shallow parsing – Probabilistic CFG, Probabilistic CYK, Probabilistic Lexicalized CFGs – Feature structures, Unification of feature structures.

UNIT IV: Representing Meaning **9 Hours**

Representing Meaning – Semantic Analysis - Lexical semantics –Word-sense disambiguation - Supervised – Dictionary based and Unsupervised Approaches - Compositional semantics Semantic Role Labeling and Semantic Parsing – Discourse Analysis

UNIT V: Discourse Analysis and Lexical Resources **9 Hours**

Discourse Analysis and Lexical Resources: Discourse segmentation, Coherence – Reference Phenomena, Anaphora Resolution using Hobbs and Centering Algorithm – Coreference Resolution – Resources: Porter Stemmer, Lemmatizer, Penn Treebank, Brills Tagger, WordNet, PropBank, FrameNet, Brown Corpus, British National Corpus (BNC)

Total: 45 Hours

References

1. "Language Processing with Java and LingPipe", by Breck Baldwin, Atlantic Publisher, 2015.
2. "Natural Language Processing with Java", by Richard M Reese, OReilly Media, 2015.
3. "Handbook of Natural Language Processing", by Nitin Indurkhya and Fred J. Damerau, 2nd Edition, Chapman and Hall/CRC Press, 2010.
4. "Natural Language Processing and Information Retrieval", by Tanveer Siddiqui, U.S. Tiwary, Oxford University Press, 2008.

List of e-Learning Resources:

1. https://www.geeksforgeeks.org/natural-language-processing-overview/

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**