SAI PALI GROUP

SAI PALI

INSTITUTE OF

TECHNOLOGY & MANAGEMENT

[FINAL COPY]

Certificate in Software Engineering With Artificial intelligence (AI)

(Two years course)

SUBMITTED TO UGANDA BUSINESS AND TECHNICAL EXAMINATIONS BOARD (UBTEB)

PREAMBLE:

Certificate Programs are widely accepted by many Higher Education institutions as an entry bridging component. In order to assist prospective students who do not have 'A' level (or did not reach the requisite grades at 'A' level, note that this option is not open to Ugandans, a Mature-Age Examination Entry Certificate, the IB qualification, or expatriate students from neighboring countries who did not achieve a C++ in the school-leaving examination in their home country, the foundation program will bridge the gap between school and university.
This innovative program is carefully formulated between the support skills needed for academic life and academic subjects needed to sustain university studies. The mix of courses offered on the program will set prospective students on a path to achieving their goals of obtaining a university education.

RATIONALE:

The Certificate in Software Engineering with Artificial Intelligence (AI) Program at Saipali Institute of Technology and Management is designed for The Primary goal of **Certificate in Software Engineering with Artificial Intelligence (AI)** is to prepare students to become competent computer programmers and software engineers who will be successful in a wide variety of career paths.

This Certification program focuses on core areas.

- **Web Designing and Development**
- **Interactive web application development**
- **System software development**
- **Application Software Development**
- **Database Management System**
- **Mobile Application development**
- **Introduction to Generative Artificial Intelligence Tools And Technique**

OBJECTIVE:

The objectives of the program are to:

1. Develop professionals with skills for Web Designing and Development
2. Develop skills for Interactive web application development
3. Develop skills for System software development
4. Develop skills for Application Software Development
5. Develop skills for Database administration.
6. Develop professionals with practical skills in Computer.
7. Build capacity with a practical orientation needed to link software engineering with Government and Industry under the broader perspective of Information and Communication Technology (ICT)
8. Build skilled ICT professionals for Uganda.
9. Develop the knowledge for generative Artificial Intelligence Tools and Technique.

LEARNING OUTCOMES

By the end of this course, the student will become:

- Software Programmer
- Software Developer
- Mobile Application developer
- Web Designer
- Web Developer
- Database Administrator
- Systems Analyst

Graduates may find employment in a wide range of industries including:

- Security and defense
- Software Companies
- Web Hosting
- Education
- Aerospace
- Health,
- Business
- Financial institution
- Communications firms,

- Private and public sector banks
- Software publishers

We expect our graduates to have mastery of crucial programming skills and to have acquired deep knowledge in fundamental areas of Computer Science (e.g., programming languages, computer architecture, databases, operating systems and networks) and to be able to apply those skills and that knowledge in various problem-solving activities, working individually and/or in teams.

We expect our students are able to integrate and synthesize theoretical and practical knowledge, perspective and critical think skills acquired through their study and have built a solid foundation for advanced Diploma which will be equivalent to Degree Program of any University. The Students of Certificate in Software Engineering with Artificial Intelligence (AI) will:

- Have a mastery of fundamental computer system principles, demonstrate proficiency in theoretical and practical aspects on various computer science fields and have built solid foundations to pursue advanced degrees.
- Have a deep understanding on design and develop computing systems and applications, and have strong programming skills and software engineering knowledge. Be capable of problem analysis, identification and definition, and able to apply mathematical foundations in the modeling and designing of its solutions.
- Have critical thinking skills, and able to design, implement and evaluate computer applications independently and/or with the teams. Have strong communication skills and function effectively on multidiscipline teams to achieve the goal.

In addition, they will be competent to work in one of the following areas:

Web Development: Design and implementation of software, including being able to perform the following: Software Development Life Cycle (SDLC).

Application Development: Devising effective and innovative ways to use computers, from robotics to Web applications to modeling biological systems to embedded systems and mobile networking. This involves integration of existing systems more than development of new programs, requiring ability to perform the following:

- Determine requirements for systems in an application area such as databases, networks, robotics, or others

- Design an integrated system to implement a system they have designed, demonstrating an understanding of trade-offs involved in the design

Database Administrator: The job of Database Administrator/System Analyst was established for the purpose/s of analyzing departmental and individual requirements for database solutions; developing, implementing and maintaining applications; documenting system components and user instructions; and serving as a technical advisor to staff and other District personnel.

Software Engineer Careers

Software engineering and computer programming are functions in high demand with unique skill sets that are increasingly important in the Internet age. Companies that are expanding their IT departments are demanding more qualified software engineers, as are technologically focused companies who depend on software engineers for their core products. Students of Software Engineering training may choose to specialize in user requirements, cost constraints, timely delivery of a product, and quality of the software product, reliability, safety or long-term maintenance.

DURATION:

The duration of the program shall be **two academic years consisting of four semesters**. Each semester lasts seventeen (17) weeks two of which are for examinations.

ENTRY REQUIREMENTS & **ELIGIBILITY:**

Uganda Certificate of Education (UCE) with at least 3 credits obtained at the same sitting / its equivalent as specified & equated by Uganda National Examination Board (UNEB) Have Year 11 (Form 5) Mathematics (or equivalent)

PASS AN INTERVIEW

Applicants whose first language is not English will be interviewed to assess their level of English competency and may be required to complete a placement indicator test.

WEIGHTING

The weighting unit is the Credit Unit (CU). The Credit Unit is a contact hour per week per semester. A contact hour is equal to (i) one lecture hour (LH), (ii) two practical hours (PH) or (iii) two tutorial hours (TH)

ASSESSMENT

Assessment is to be done by progressive assessments (like tests, assignments, group work) during the semester and final examination. The final examination may be purely written, purely practical or having a written and practical component. Progressive assessments constitute 40% of the final score and the final examination will constitute 60%.

GRADING SYSTEM:

Assigned grades								
Marks	80-100	75-79	70-74	65-69	60-64	55-59	50-54	0-49
Alpha Grade	**A**	**B+**	**B**	**C+**	**C**	**D+**	**D**	**F**
Grade Point	**5**	**4.5**	**4**	**3.5**	**3**	**2.5**	**2.0**	**0**

Classification of awards for diploma:	
Class	**CGPA**
Class I (Distinction)	4.4-5.00
Class II (Credit)	2.80-4.39
Class III (Pass)	2.00-2.79

ACADEMIC STAFF:

The list of staff members in the Department of Software Engineering is as follows:

NAME	RANK
Mr. Vaibhav Sonaje	**Sr. Lecturer**
Mr. Shaukat Ali	**Sr. Lecturer**
Ms. Punam Dhabale	**Sr. Lecturer**
Mr. Taquee Ansari	**Sr. Lecturer**
Mr. Katongole Mark Anold	**Sr. Lecturer**

CERTIFICATE IN SOFTWARE ENGINEERING WITH ARTIFICIAL INTELLIGENCE (AI)

PROGRAM CODE: CSE

YEAR I SEMESTER I				
MODULE NAME	**MODULE CODE**	**PAPER CODE**	**SUB MODULE**	**CREDIT UNITS**
FOUNDATIONS OF SOFTWARE DEVELOPMENT	CSE - M I		**CORE UNITS**	
		CSE 111	Elementary Programming Using C	4
		CSE 112	Website Technologies	4
			SUPPORTING UNITS	
		CSF 113	Computational Set Theory and Functions (Mathematics)	3
		MOA 114	Microsoft Office Application	3
		BCS 115	Basic Communication skills	3
		CSE 116	Real life project I [**static web site**]	3
YEAR I SEMESTER II				
MODULE NAME	**MODULE CODE**	**PAPER CODE**	**SUB MODULE**	**CREDIT UNITS**
OBJECT-ORIENTED PROGRAMMING LANGUAGES	CSE - M II		**CORE UNITS**	
		CSE 121	Fundamentals of Object-Oriented programming using C++	4
		CSE 122	Database Programming	4
		CSE 123	Web Development in PHP-I	4

			SUPPORTING UNITS	
		CSE 124	Introduction to Computer Hardware	3
		MRG 125	Mathematical Reasoning and Geometry (Mathematics)	3
		CSE 126	Real life project II [**Dynamic web site**]	3
		CIT 120	Industrial Training I	2

YEAR II SEMESTER I				
MODULE NAME	**MODULE CODE**	**PAPER CODE**	**SUB MODULE**	**CREDIT UNITS**
MOBILE APPLICATION TECHNOLOGIES	CSE - M III		**CORE UNITS**	
		CSE 211	Fundamentals of programming in Java	4
		CSE 212	Mobile Application Development	4
			SUPPORTING UNITS	
		CSE 213	Basics of Networking	3
		CSE 214	Software Engineering Principles	3
		ISP 215	Introduction to Statistics and Probability (Mathematics)	3
		CSE 216	Real Life Project III **[Mobile application]**	3
YEAR II SEMESTER II				
MODULE NAME	**MODULE CODE**	**PAPER CODE**	**SUB MODULE**	**CREDIT UNITS**

FOUNDATIONS OF DATA SCIENCE AND ARTIFICIAL INTELLIGENCE	CSE - M IV		**CORE UNITS**	
		GAI 221	Generative Artificial Intelligence	4
		CSE 222	Introduction to Python	4
		CSE 223	Web development in PHP-II	4
			SUPPORTING UNITS	
		LAC 224	Linear Algebra and Calculus (Mathematics)	3
		PES 225	Principles in Entrepreneurship skills	3
		CSE 226	Real Life Project IV **[Web Development in PHP using Python/AI tools]**	3
		CIT 220	Industrial Training II	2

SEMESTER I

MODULE NAME: FOUNDATION OF SOFTWARE DEVELOPMENT

MODULE CODE: CSE – M I

1. COURSE NAME: (CSE 111) ELEMENTRY PROGRAMMING USING C

LEVEL: YEAR ONE, SEMESTER-I

CREDIT UNITS: 4

CONTACT HOURS: 45

COURSE DESCRIPTION:

C is a high-level and general purpose procedure oriented programming language that is ideal for developing system applications. Originally intended for writing system software, the course provides a strong base in the principles and practice of structured programming. A high level programming language such as C is used to explain the principles of programming and provide students with hands on practical skills. Areas covered include program structure, data structures, syntactical and semantic correctness, planning and segmentation in programming as well as working with data files.

COURSE OBJECTIVES:

The aims of this course are, to:

1. Knowledge about the various programming languages
2. Knowledge in basic programming concepts
3. Comprehensive knowledge about structured programming
4. Skills in planning and organization of programming projects
5. Techniques of evaluating syntactic and semantic correctness of a computer program
6. Strong practical foundation in programming

LEARNING OUTCOMES:

Upon successful completion of the course, students shall be able to:

1. Explain the key differences between the various programming languages
2. Demonstrate understanding about the basic programming concepts
3. Build software using a programming language such as C
4. Plan and organize a programming project
5. Evaluate a computer program for syntactic and semantic correctness

COURSE OUTLINE:

UNIT NO.	**DESCRIPTION**	**No. of LECTURES**
1	**1. Programming Language As Tools** 1.1. Machine language 1.2. Assembly language 1.3 High level languages 1.4. Compilers and Interpreters	**2**
2	**2. Problem Solving Concept:** 2.1. Requirement of solving problems by computer 2.2. Problem solving aspects 2.3. Algorithm And Flowchart 2.4. Definition & Characteristics of algorithm 2.5. Simple examples on algorithms 2.6. Flow charts	**6**
3	**3. Arithmetic problem solving using algorithm and flow charts:** 3.1. Examples on Simple Arithmetic Statements, Conditional Statement & Iterative Statements (such as Addition/Multiplication, check number is positive/negative, Maximum of 2 numbers & 3 numbers, sum of first n numbers, sum of given n numbers, reverse digits of number check number is palindrome, check number is prime, factorial of number, factors of Number, etc.)	**9**
4	**4. Introduction to C Language** 4.1. Introduction to C	**3**

	4.2. Features of C 4.3. Application of C programming Language. 4.4. Structure of C program	
5	**5. Tokens** 5.1. C Character Set, Identifiers and Keywords 5.2. Variables ,literals and constants 5.3. Data types- Basic data types, Enumerated types, 5.4. Type casting 5.5. Declarations, Expressions	**5**
6	**6. Operators and Expressions** 6.1. Unary Operators 6.2. Binary operators 6.3. Arithmetic Operator 6.4. Relational operators 6.5. Bit wise operators 6.6. Assignment operators 6.7. Logical Operator 6.8. Increment And Decrement Operator 6.8. Ternary conditional operator 6.9. Operator Precedence and Associativity	**6**
7	**7. Formatted And Unformatted Input/output functions in C** 7.1. printf, scanf functions 7.2. getchar, putchar, getch,getche,putche,putch functions 7.3. gets, puts functions 7.4. Escape sequence characters 7.5. Format specifiers	**3**
8	**8. Control Statements** 8.1. if, if - else Statements 8.2. Nested if Statements 8.3. else if ladder, switch statement, 8.4. Loop (while, do…while, for) 8.5. break, continue, goto statements.	**4**
9	**9. Functions** 9.1. Introduction to Functions 9.2. Function Arguments 9.3. Library & User defined functions	**4**

	9.4. Methods of Calling Function 9.5. Recursion 9.6. Storage Classes	
10	**10. Arrays** 10.1. Introduction 10.2. Array Declarations 10.3. Single dimension Arrays 10.4. Two dimension Arrays 10.5. Arrays & Function	**4**
11	**11.Pointers** 11.1. Pointer declaration, initialization 11.2. Dereferencing pointers 11.3. Pointer arithmetic 11.4. Pointer to pointer 11.5. Arrays and pointers 11.6. Functions and pointers – passing pointers to functions, function returning pointers 11.7. Dangling, void, Null And Wild Pointer 11.8. Dynamic memory allocation	**4**
12	**12. Strings** 12.1. Declaration and initialization, format specifies 12.2. Standard library functions 12.3. Strings and pointers 12.4. Array of strings 12.5. Command Line Arguments	**4**
13	**13.Structures and Unions** 13.1. Creating structures 13.2. Accessing structure members (dot Operator) 13.3. Structure initialization 13.4. Array of structures 13.5. Passing structures to functions 13.6. Nested structures 13.7. Pointers and structures 13.8. Unions 13.9. Difference between structures and unions 13.10. typedef in C	**4**

14	**14. File Handling** 14.1. Types of Files 14.2. Operations on files 14.3. Random access to files 14.4. C program to read/write and delete operation to/from file	**2**
15	**15. C Preprocessor** 15.1. Format of Preprocessor directive 15.2. File Inclusion directive 15.3. Macro substitution, nested macro, argumented macro	**2**

Reference Books:

1. Introduction to algorithms – Cormen, Leiserson, Rivest, Stein
2. The C Programming Language, Brian W. Kernighan, Dennis M. Ritchie,ISBN:9788120305960, PHI Learning
3. How to Solve it by Computer, R.G. Dromey, ISBN: 9788131705629, PearsonEducation
4. A Structured Programming Approach Using C, Behrouz A. Forouzan, RichardF. Gilberg ISBN:9788131500941, Cengage Learning India
5. Programming in ANSI C, E. Balaguruswamy, ISBN: 9781259004612, Tata Mc-Graw Hill Publishing Co Ltd.-New Delhi

MODE OF DELIVERY:

Lectures, Practical's, Assignment.

COURSE ASSESSMENT:

Assessment	Practical Work	Examinations	Total
Contribution	40%	60%	100%

2. COURSE NAME: (CSE 112) WEB TECHNOLOGIES

LEVEL: YEAR ONE, SEMESTER-I

CREDIT UNITS: 4

CONTACT HOURS: 45

COURSE DESCRIPTION:

This course covers the design, implementation and testing of web-based applications including related software, databases, interfaces and digital media.

Web development broadly refers to the tasks associated with developing websites for hosting via intranet or Internet. The Web development process includes Web design, Web content development, client-side/server-side scripting and network security configuration, among other. Web development is also known as website development. Web development is the coding or programming that enables website functionality, per the owner's requirements. It mainly deals with the non-design aspect of building websites, which includes coding and writing markup. Web development ranges from creating plain text pages to complex Web-based applications, social network applications and electronic business applications.

COURSE OBJECTIVES:

The objectives of the course are to:

1. Introduce students to the design, implementation and testing of web-bases application.

2. Introduce students to various Internet technologies such HTML, HTTP protocols, W3C standards, etc.

3. Introduce students to client-side and server-side scripting programming.

4. Introduce students to web site development.

5. Introduce students to mobile Internet, and design and development of standard compliant mobile web.

LEARNING OUTCOMES:

Upon successful completion of the course, students will be able to:

1. Describe the structure of the World Wide Web as interconnected hypertext documents.

2. Describe the importance of the HTTP protocol in Web applications.

3. Design Static Website.

4. Demonstrate an understanding of XML syntax and show how to display such documents in Web applications.

COURSE OUTLINE:

UNIT NO.	DESCRIPTION	No. of LECTURES
1	**1. HTML Introduction:** 1.1. What is HTML5? History and Major Actors, Little Retrospective, What Is The W3C? 1.2. Structure of a Web Page, Editors (Notepad++, VScode, Sublime, Atom) 1.3. HTML Elements, Tags, Text, Formatting, Pre, Attributes, Text 1.4. Links, Comments, Fonts 1.5. Tables 1.6. Forms, Input, Text Fields , Password, Reset, Submit, Checkboxes, Radio, Select, Hidden Fields, Upload, Text-areas, Special Tags, Body, Meta, Style, 1.7. Div, Layouts, Entities, Scripts, Formatting Tags, Bold, Paragraphs, Headings, Line Breaks, Horizontal Rule, Email, Italic, Code, Superscript, Subscript, Strikethrough, HTML Colors	4
2	**2. HTML5** : 2.1. What is HTML5, What Is The WHATWG?, Vision And Philosophy Behind HTML5, Compatibility, Utility, Interoperability, Universal Access, Future Of HTML5, Getting Started With HTML5, The State Of Browser Support, Feature Detection, Support For Legacy Browsers, Graceful Degradation, Emulation, Developer Tools 2.2. HTML5 DOCTYPE: Page Encoding, HTML5 Markup, New and Updated Elements, Structural Elements, New Attributes,	10

	Deprecated Elements and Attributes, HTML5 And CSS3, Browser Support 2.3. Forms: What Are The Needs For Web Applications?, Current Solutions, New Input Types, New Attributes, Form Validation, 2.4.Browser Support. Audio and Video: The State of Web Audio and Video Based On Plug-in, The State Of Audio And Video Codec, Video/Audio Codec And Browser Support 2.5. HTML5 Canvas: Overview Of Graphics In The Browser	
3	**3. CSS (Cascading Style Sheet)** 3.1. CSS Introduction, CSS Syntax 3.2. CSS Selectors (Ids, Classes, universal, child, descedent), How it works 3.3.CSS Styling, Styling , Backgrounds, Styling Text, Styling Fonts, Styling Links, Styling Lists, Styling Tables 3.4. CSS Box Model, CSS Border, CSS Outline, CSS Margin, CSS Padding, CSS Advanced, CSS Grouping/Nesting, CSS Dimension, CSS Display, CSS Positioning, CSS Floating, CSS Align, CSS Pseudo-class, CSS Pseudo-element, 3.5. CSS Navigation Bar 3.6. CSS Image Gallery, CSS Image Opacity , CSS Image Sprites, CSS Media Types, CSS Attribute Selectors	**7**
4	**4. JavaScript**: 4.1. JavaScript Introduction, What is JavaScript? Pre Requirement 4.2. Creating a JavaScript file, Tools Required, Basics of JavaScript 4.3.Structure of a JavaScript Code 4.4. Data Types, Variables & Operators, Looping & Conditional Statements 4.5. Functions Types & Objects, Arrays, Numbers, Strings, Dates, Objects	**10**
5	**5.DOM (Document object model):** 5.1. What is DOM? Element Nodes, Getting Element Nodes, Changing DOM Content, Creating DOM Elements, Events & Listeners, Events Overview, Responding to Mouse Events, Form Events, DOM Manipulation, Using Firebug	**4**
6	**6. Introduction to Jquery** Traversing the DOM, Working with the DOM, Listening to DOM Events Styling, Responsive web design & resources, The Concept Of Responsive Web Design Adjusting Screen Resolution, CSS3	**10**

	MEDIA QUERIES, Flexible Images, Showing or Hiding Content, Responsive web design resources	
7	**7. Basics of Bootstrap** 7.1. Describe Bootstrap 7.2. Describe the process of setting up Bootstrap 7.3.Outline the File structure and the Default source code in Bootstrap 7.4. Use Bootstrap with CSS and JS 7.5. Describe the significance of Bootstrap templates 7.6. Describe how strings can be used as function arguments 7.7. Describe the significance of Bootstrap template	

Reference Books

1. Textbook of Web Design – Joel Sklar, Cengage Learning

2. HTML: The Complete Reference – Thomas A. Powel

3. Web Technologies – Uttam K. Roy, Oxford

MODE OF DELIVERY:

Lectures, Practical, Tutorials, Group work, Individual studies

COURSE ASSESSMENT:

Assessment	Practical	Examination	Total
Contribution	40%	60%	100%

3. COURSE NAME: (CSF 113): COMPUTATIONAL SET THEORY AND FUNCTIONS (Mathematics)

LEVEL: YEAR ONE, SEMESTER-I

CREDIT UNITS: 3

CONTACT HOURS: 45

COURSE OBJECTIVES:

The objectives of the course are to:
1. To teach students Algebraic Expressions
2. To teach students Equations and Inequalities.
3. To teach students Discrete Structures
4. To teach students Polynomials and rational functions.
5. To teach Students Exponential and Logarithmic Functions.

COURSE DESCRIPTION:

Basic mathematics involves mathematical research in mathematics as well as in areas of science where basic math plays a central and essential role and emphasizes algorithms, Discrete Structures and Polynomials and Rational Functions.

LEARNING OUTCOME

By the end of this module the students should be able to incorporate the knowledge acquired here to:

i. Utilize set operations to solve computational problems.
ii. Implement algorithms that use set theory concepts to manage data structures efficiently.
iii. Enhance problem-solving skills by applying set theory to various computational scenarios.
iv. Analyze the efficiency of set operations and their impact on algorithm performance.
v. Apply set theory concepts to query databases and manage data relationships.
vi. Design efficient database schemas using set theory principles.
vii. Comprehend the definition and purpose of functions in C programming.
viii. Learn about function declaration, definition, and invocation.
ix. Develop skills to design reusable and modular code through functions.
x. Implement various types of functions, including user-defined and standard library functions.
xi. Understand different methods of passing parameters (by value and by reference).
xii. Grasp the concepts of scope and lifetime of variables within functions.
xiii. Enhance code readability, maintainability, and reusability through function usage.
xiv. Apply functions to break down complex problems into manageable pieces.
xv. Understand the principles of web design, including layout, color theory, typography, and user experience.

xvi. Learn HTML, CSS, and JavaScript for creating and styling web pages.
xvii. Develop skills to create responsive web designs that work on various devices and screen sizes.
xviii. Use frameworks like Bootstrap to facilitate responsive design implementation.
xix. Implement dynamic content and interactive features using JavaScript.
xx. Learn about DOM manipulation and event handling to enhance user interaction.
xxi. Utilize web development tools such as version control (Git), code editors (VS Code), and debugging tools.
xxii. Follow best practices for web development, including accessibility, SEO, and performance optimization.
xxiii. Understand the basics of server-side scripting and databases to integrate with the frontend.
xxiv. Learn about AJAX and RESTful APIs to facilitate data exchange between client and server.
xxv. Combined Learning Outcomes
xxvi. Apply set theory concepts to optimize algorithms and data handling in web applications.
xxvii. Use functions in C programming to build efficient backend services for web applications.
xxviii. Develop a robust approach to solving complex problems by integrating computational theories and practical programming skills.
xxix. Improve logical thinking and analytical abilities by applying theoretical knowledge to real-world scenarios.

COURSE CONTENT

UNIT NO.	DESCRIPTION	No. of LECTURES
1	**1. Number Bases** 1.1. Addition of numbers in the same base. 1.2. Conversion of numbers from one Base to another, restricted to only Binary Base, Octal Base, Decimal Base and Hexadecimal Base.	5
2	**2. Indices And Logarithms** 2.1. Simplifying simple expressions involving basic laws of positive and negative indices of integral and rational numbers. 2.2. Simplifying expressions involving laws of logarithms. 2.3. Problems involving changing the base of logarithmic expressions. 2.4. Equations involving indices. Importance placed on the binary logarithms and common logarithms (that is, to base ten)	10

3	**3. Set Theory** 3.1. Sets and their representation. 3.2. The empty set, finite set and infinite set. 3.3. Equal sets. 3.4. The power set. 3.5. Subsets 3.6. Universal set. 3.7. Venn diagrams, extended three. 3.8. Union and intersection of sets. 3.9. Difference of sets. 3.10. Compliment of sets	10
4	**4. Functions** 4.1. Pictorial representation of a function, the domain, co-domain and range of a function. 4.2. Identifying one on one, onto and surjective functions. Composition of functions up to only two. Inverse of functions.	5

Reference Books

1. Bird J.O. and May A.J.C. (1994) Technician Mathematics. Volume 3 Longman Scientific and Technical Glyn, J. (2015)
2. Modern Engineering Mathematics 5th edn Pearson Education Limited. Stroud
3. K.A. (2013) Engineering Mathematics 7th edn UK Macmillan Education

MODE OF DELIVERY:
Lectures, Practical, Tutorials, Group work, Individual studies

COURSE ASSESSMENT:

Assessment	Tests	Examination	Total
Contribution	40%	60%	100%

4. COURSE NAME: (MOA 114) MICROSOFT OFFICE APPLICATION

LEVEL: YEAR ONE, SEMESTER-I

CREDIT UNITS: 3

CONTACT HOURS: 45

COURSE DESCRIPTION:

To provide application packages, internet and intranet tools, web hosting. Essential for a modern office for day-to-day office management and e-governance.

COURSE OBJECTIVES:

By the end of this course, the student should be able to:
1. To Teach student exchange of information.
2. To enable learners, manage administrative documents.
3. To train learners how to handle numerical data.
4. To train learners how to create presentation.
5. To train learners how to create spreadsheets.
6. To equip students with knowledge and skills in meeting planning and management of work schedules.

LEARNING OUTCOMES:

By the end of this course, the student will be able to:
1. To demonstrate the various tools of MS Office Application.
2. To design professional word documents and spread sheets.
3. To demonstrate how to professionally work with databases. Design world class presentation for targeted audience.

COURSE OUTLINE:

UNIT NO.	**DESCRIPTION**	**No. of LECTURES**
1	**1. Working with word processing and advanced word processing** 1.1. Introduction to Microsoft word 1.2. Working with documents and document views	5

	1.3. Editing and Formatting text 1.4. Formatting and indenting paragraphs 1.5. Character and Paragraph styles 1.6. Applying Bullets and numberings 1.7. Checking spelling and grammer 1.8. Using Find and replace 1.9. Applying Headers and footers 1.10. Working with tabs and tables 1.11. Inserting graphics 1.12. Working with Mail merge	
2	**2. Spreadsheets and advanced spreadsheets** 2.1. Introduction to Microsoft Excel 2.2. Basic workbook skills 2.3. Working with Selection in Excel 2.4. Working with columns and rows 2.5. Applying Number formats 2.6. Editing and Formatting of text 2.7. Cell formatting in Excel 2.8. Working with tables 2.9. Using of formulas 2.10. Using the cut, copy and paste commands 2.11. Data Management 2.12. Creating charts 2.13. Using page set up	10
3	**3. Creating presentations and advanced presentations** 3.1. Introduction to Microsoft PowerPoint 3.2. Using basic presentation skills 3.3. Working with presentations 3.4. Editing and proofing text 3.5. Formatting bullets and Numbers 3.6. Using Tables 3.7. Using Graphic images, SmartArt and Drawn objects 3.8. Working with slides 3.9. Creating basic charts 3.10. Adding special effects 3.11. Using slide show view	5
4	**4. Databases and Advanced databases** 4.1. Introduction to Microsoft Access 4.2. Creating tables in Acccess 4.3. Setting field properties 4.4. Working with tables 4.5. Editing tables 4.6. Finding and Filtering data	10

	4.7. Using Simple Queries 4.8. Modifying query results 4.9. Using Operators in Queries 4.10. Creating relationships 4.11. Creating basic reports 4.12. Using advanced report design	

Reference Books:

1. Enhanced Microsoft Office 2013: Introductory (Microsoft Office 2013 Enhanced Editions) 1st Edition by Misty E. Vermaat (Author)
2. The handbook of office automation Ralph Thomas Reilly.
3. Office Automation – Andrew Doswell

MODE OF DELIVERY:

Lectures, peer discussions, demonstrations

COURSE ASSESSMENT:

Assessment	Assignment	Test	Final Exam	Total
Contribution	10%	30%	60 %	100 %

5. COURSE NAME: (BCS 115) BASIC COMMUNICATION SKILLS

LEVEL: YEAR ONE, SEMESTER-I

CREDIT UNITS: 3

CONTACT HOURS: 45

COURSE DESCRIPTION:

This program is designed to develop the knowledge and skills you need to succeed as a technical communications professional. You will learn to write and present information that is clear, concise, and audience-focused.

The program focuses on specialized communication formats including proposal writing, online documentation and technical manuals.

COURSE OBJECTIVES:

1. To enable students understand the genre and manipulate the structure of selected technical documents.
2. To enable students convey clearly, cogently and correctly, through written media, the technical aspects of a practice to non- specialist audiences.
3. To teach students recognize and use the rhetorical and stylistic elements necessary for the successful practice of scientific and technical communication.
4. To teach students appreciate their obligations as prospective practitioners in their chosen field to the laypersons affected by your work.
5. To teach students basics of Oral-communication
6. To teach students basics of Non-verbal communication
7. To teach students basics of Listening

LEARNING OUTCOMES:

A student who successfully completes the course will be able to: -
1. Apply an appropriate communication strategy for an audience or reader.
2. Define technical writing and its uses Analyze audience role in technical writing.
3. Demonstrate the shorter writing tasks: email Write persuasively and connect with readers.

COURSE OUTLINE:

UNIT NO.	DESCRIPTION	No. of LECTURES

1	**1. INTRODUCTION TO COMMUNICATION :** 1.1. Definition 1.2. Roles/purpose of communication 1.3. Principles of communication 1.4. Role of ict in communication	3
2	**2. COMMUNICATION PROCESS** 2.1. Barriers of communications 2.2. Solutions to barriers of communication	2
3	**3. CLASSIFICATIONS OF COMMUICATION** 3.1. Internal communication 3.2. External communication 3.3. Formal communication 3.4. Informal communication	4
4	**4. FORMS/TYPES OF COMMUNICATION** **4.1. Verbal communication** o Types of verbal communication o Importance of verbal communication o Challenges of verbal communication **4.2. Non-verbal communication** o Types of non- verbal communication o Importance of non-verbal communication o Challenges of verbal communication **4.3. Written communication** o Types of written communication o Importance of written communication o Challenges of written communication	5

5	**5. CHANNELS OF COMMUNICATION** 5.1. Importance of channels of communication 5.2. Factors to consider when choosing channels of communication	2
6	**6. WRITING SKILLS** **6.1.Types of writing** A. **Functional writing** o Business letters i.e letter of inquiry,appolgy,complaint,appreciation etc. o Memos o Circulars o Public notices o Curriculum vitea o Electronic mails o Summary writing	2
7	**7. REPORT WRITING SKILLS** 7.1. Types of reports 7.2. Format of report writing 7.3. Qualities of a good report	3
8	**8. CONDUCTING MEETINGS AND MINUTE WRITING** 8.1. Structure of minute writing 8.2. Terminologies in minute writing 8.3. Roles or purposes of minute writing 8.4. Types of meetings 8.5. Challenges in conduct of meetings and minute writing 8.6. Advantages of meetings 8.7. Disadvantages of meetings	3

9	**9. INTERVIEWS** 9.1. Types of interviews 9.2. Roles of interviewers and interviews 9.3. Challenges of interviews	2
10.	**10. LISTENING SKILLS** 10.1.Types of listening 10.2. Process ofeffective listening 10.3. Elements of good listerners 10.4. Barriers of listening external and internal	2
11	**11. BUSINESS COMMUNICATION ETTIQUTES**	1

REFERENCE BOOKS

1. J. Technical Writing and Communication, E.F. Bloomhower

2. Technical Communication, V.1.0., Mike Markel, Eight Edition

MODE OF DELIVERY:

Lectures, Case studies, Practical's, peer discussions and Group work

COURSE ASSESSMENT:

Assessment	Practical	Final Exam	Total
Contribution	40%	60 %	100 %

6. COURSE NAME: (CSE116): REAL LIFE PROJECT I [STATIC WEBSITE]

LEVEL: YEAR ONE, SEMESTER-I

CREDIT UNITS: 3

CONTACT HOURS: 45

COURSE OBJECTIVES:
The objectives of the course are to:

1. To introduce students to the field of website development using HTML5.
2. To teach students how to enhance their analyzing and problem solving skills.
3. To use their skills while developing websites using HTML5.
4. To teach students how to develop Universal Stationary websites.
5. To teach students how to develop different static website.

COURSE DESCRIPTION:

The purpose of this course is to introduce students an in-depth training of HTML5. This program will provide them practical knowledge of building both static and dynamic websites. Websites are an increasingly important aspect of modern business, so we begin with the relation between websites and the strategic goals of the organization. We move on to discuss the technical, cultural and interpersonal skills necessary to successfully manage websites from start to finish. The course emphasizes that website development is a professional discipline with its own tools, body of knowledge and skills. Concepts are reinforced by case studies covering a wide variety of website projects

LEARNING OUTCOME:

Upon successful completion of the course, students will be able to:
1. Apply HTML5 concepts in problem solving.
2. Develop programming in HTML5 and related technologies like bootstrap.
3. Write diversified solutions using HTML5.

COURSE OUTLINE:
Students can develop a website on any of the following according to their supervisor:
1. Universal Stationary website
2. University Stationary website
3. Static Hospital website
4. Static school Website
5. Food Vending Website
6. Beauty Care Center Website
7. Static Perfumes Website

8. Static Game website
9. Sports Club Website
OR
Identify suitable project as per the guidelines of the institution and the project guide.

MODE OF DELIVERY

Presentations, Project supervision

COURSE ASSESSMENT

1.	Project Proposal	20%
2.	Project Development	40%
3.	Project Report	20%
4.	Presentation	20%

SEMESTER II

MODULE NAME: OBJECT ORIENTED PROGRAMMING LANGUAGE

MODULE CODE: CSE – M II

1. COURSE NAME: (CSE 121) FUNDAMENTALS OF OBJECT ORIENTED PROGRAMMING USING C++

LEVEL: YEAR ONE, SEMESTER-II

CREDIT UNITS: 4

CONTACT HOURS: 45

COURSE DESCRIPTION:

The object-oriented programming paradigm has been used in many projects to build complex software systems. This course provides students with the opportunity to acquire skills for building software based on the object oriented programming paradigm. C++ and/or Java are some of the languages used in this course.

COURSE OBJECTIVES:

The course aims to enable students to:

1. Understand and use the Object Orientation Paradigm for software development
2. To understand how to develop software using a programming language like C++. /java
3. Understand the use of basic object-orient programming language features in a working program
4. Understand the use of the following advanced features of inheritance, encapsulation, overloading, polymorphism, abstract classes and interfaces in a program

LEARNING OUTCOMES:

Upon successful completion of the course, students should be able to:

1. Demonstrate the understanding and application of Object oriented design techniques for software development using a programming language like C++ and/ or Java

2. Correctly use the basic features in a working program: objects, classes, methods, IO handling, decisions and iterations

3. Apply and interpret the following advanced features in a working program: inheritance, encapsulation, overloading, polymorphism, abstract classes and interfaces

COURSE OUTLINE:

Unit No	**Contents**	**No Of Lectures**
1	**1. Introduction to C++** 1.1. Basics of C++, 1.2. Structure of C++ Program, keywords in C++, 1.3. Data types hierarchy in C++, 1.4. Operators in C++: Scope resolution operator, Insertion and Extraction operator New and Delete operators. 1.5. Reference variable. 1.6. Manipulators function: endl, setw, set fill, set precision.	**6**
2	**2. Object oriented Concepts** 2.1. Object oriented concepts 2.2. Features, 2.3. Advantages and Applications of OOP 2.4. Difference between Procedure oriented programming and object Oriented programming.	**2**
3	**3. Classes and Objects** 3.1. Structure sand class, Class, Object, Access specifies, 3.2. Class members, 3.3. Defining member functions :Inside and outside the class definition, 3.4. Creating objects. String class, operation on string, Array of objects. 3.5. 'this' pointer.	**5**
4	**4. Function in C++** 4.1. Call by reference, Return by reference, 4.2. Function overloading and default arguments 4.3. Inline function 4.4. Passing and returning objects from functions, Static class members	**4**

	4.5. Friend Concept – Function, Class	
5	**5. Constructors and Destructors** 5.1. Memory allocation and static data members 5.2. Definition of constructor Types of constructors: Default Constructor 5.3. Constructor with default arguments 5.4. Parameterized Constructor 5.5. Copy Constructor 5.6. Overloaded constructors in a class 5.7. Destructors	**4**
6	**6. Inheritance** 6.1. Introduction 6.2. Types of Inheritance: Single inheritance Multiple inheritance, Multilevel inheritance Hierarchical inheritance Hybrid inheritance. 6.3. Derived Class Constructor sand Destructors 6.4. Ambiguity in multiple Inheritances, virtual base classes, Abstract Base class.	**4**
7	**7. Virtual Function &Polymorphism** 7.1.Introduction, Pointer to object, Pointer to derived 7.2.class, Overriding member functions, Virtual function, Rules for virtual functions, pure virtual function, Run- time type information (RTTI)	**4**

REFERENCE BOOKS:

1. Object Oriented Programming with C++ by Robert Lafore
2. Object Oriented Programming with C++ by E. Balagurusamy
3. Object Oriented Modeling and Design by James Rumbough
4. The Complete Reference C++ by Herbert Schildt
5. let us C++ by – YashwantKanitkar
6. Mastering C++ by Venugopal, T Ravishankar, RajkumarTHM Pub.
7. Trouble free C++ by HarimohanPande, ANE publication

MODE OF DELIVERY:

Lectures, Practical, Group work

COURSE ASSESSMENT:

Assessment	Practicals	Examination	Total
Contribution	40%	60%	100%

2. COURSE NAME: (CSE 122) DATABASE PROGRAMMING

LEVEL: YEAR ONE, SEMESTER-II

CREDIT UNITS: 4

CONTACT HOURS: 45

COURSE DESCRIPTION:

This course engages students to implement database models by creating physical databases using SQL. Students will learn basic SQL syntax and the rules for constructing valid SQL statements to generate report-like output. Demonstrations and hands-on practice reinforce the fundamental concepts.

COURSE OBJECTIVES:

The objective of this course is to study the basics DBMS and to learn SQL.

LEARNING OUTCOMES:

Upon successful completion of the course, students shall be able to:

1. Student will delve into the fascinating world of database and gain essential skill related to managing and interacting with data.
2. Understand the fundamental concepts that how database work.
3. Plan and design a simple relational database system.

COURSE OUTLINE:

UNIT NO.	DESCRIPTION	No. of LECTURES
1	**1. Introduction of RDBMS** 1.1. Explain the concept of data and database 1.2. Describe the approaches to data management 1.3. Define a Database Management System (DBMS) and list its benefits 1.4. Explain the different database models 1.5. Define and Explain RDBMS 1.6. Lists the difference between a DBMS and RDBMS	5
2	**2. E-R Model and Normalization**	10

	2.1. Define and Describe data modeling 2.2. Identify and describe the components of the E-R model 2.3. Identify the relationships that can be formed between entities 2.4. Explain E-R diagram and their uses 2.5. Describe the various Normal Forms 2.6. Outline the uses of different Relational Operators	
3	**3. Structure of Relational Databases** 3.1. Concepts of a table, a row, a relation, a tuple and a key in a relational database 3.2. Integrity constraints (primary key, referential integrity, Null constraint, unique constraint, check constraint)	**5**
4	**4. SQL** 4.1. Introduction 4.2. DDL commands (create, drop, alter) with examples 4.3. Basic structure SQL query 4.4. Set operations 4.5. Aggregate functions 4.6. Null values 4.7. Nested Sub-queries 4.8. Modifications to Database (insert, delete, update) 4.9. SQL mechanisms for joining relations (inner joins, outer joins and their types) 4.10. Examples on SQL (case studies)	**15**
5	**5. Creating tables** 5.1. List SQL Server data types 5.2. Describe the procedure to create, modify and drop tables in an SQL Server database 5.3. Describe the procedure to add, modify and drop columns in a table 5.4. Normal forms (only definitions) 1NF, 2NF, 3NF, BCNF 5.5. Examples on Normalization	**5**
6	**6. Transact-SQL** 6.1. Explain Transact-SQL	**10**

	6.2. List the different categories of Transact-SQL statements 6.3. Explain the various data types supported by Transact-SQL 6.4. Explain the transact-SQL language elements 6.5. Describe the logical order of operators in the SELECT statement	

Reference Books:

1. Database System Concepts, Henry F. Korth, Abraham Silberschatz, S. Sudarshan,Tata McGraw-Hill Education
2. Database Management Systems, Raghu Ramakrishnan and Johannes Gehrke, McGraw-Hill Science/Engineering/Math; 3 Edition
3. Database Systems, Shamkant B. Navathe, RamezElmasri,Pearson Higher Education

MODE OF DELIVERY:

Lectures, Practical, Assignments

COURSE ASSESSMENT:

Assessment Contribution	Practicals	Examination	Total
	40%	60%	100%

3. COURSE NAME: (CSE 123) WEB DEVELOPMENT IN PHP-I

LEVEL: YEAR ONE, SEMESTER-II

CREDIT UNITS: 4

CONTACT HOURS: 45

COURSE DESCRIPTION:

PHP – a server-side scripting language created by Rasmus Lerdorf in 1995 – is installed on more than 200 million of web sites and more than 2 million of web servers. As it is open source by nature and an efficient programming language, it is today's one of the widely used languages for web development. Thus, there is a huge demand for PHP programmers.

COURSE OBJECTIVES:

1. To Design dynamic and interactive Web pages.
2. To Develop Dynamic Web applications.
3. To use database on website

LEARNING OUTCOMES:

1. Learn Core-PHP, Server-Side Scripting Language
2. Learn PHP-Database handling.

COURSE OUTLINE:

Unit No	**Contents**	**No Of Lectures**
1	**1. Introduction:** **The** Origin of PHP is better than Its alternatives How PHP works with the Web Server Hardware and Software requirements and installation PHP Pros and Cons PHP: past, present and future Strength of PHP	**2**
2	**2. Basic PHP Development :** How PHP scripts work Basic PHP syntax PHP variables PHP data types Displaying type information Testing for a specific data type Operators Variable	**5**

	manipulation Dynamic variables String in PHP, Conditional Statements & Looping The if statement Using the else clause with if statement, multiple if The switch statement Using the ? Operator The while statement	
3	**3. Arrays**: Single-Dimensional Arrays Multidimensional Arrays Casting Arrays Associative arrays Accessing arrays Getting the size of an array Looping through an array Looping through an associative array Examining arrays Joining arrays	**6**
4	**4. Functions**: Using Functions Create a PHP Function Adding Parameters Return Values PHP Library Function Array functions String functions Date and time functions other important functions	**6**
5	**5. OOPS - PHP Classes and Objects:** Introduction of Objects oriented programming Define a class Creating an object, Object properties Object methods constructors and destructors Class constants, Access modifier, Class inheritance Abstract classes and methods Object serialization	**8**
6	**6. Forms and Date Time:** PHP Form Handling Form Validation Super global variables The server array A script to acquire user input Importing user input Accessing user input Combine HTML and PHP code Using hidden fields Redirecting the user File upload and scripts Different Date & Time Functions Date arithmetic to get future and past date	**8**
7	**7. Cookies and Session :** The anatomy of a cookie Setting a cookie with PHP Deleting a cookie Creating session cookie Working with the query string Creating query string What is session Starting a session Working with session variables Destroying session Passing session Ids Encoding and decoding session variables	**10**
8	**8. Working With the File System:** Creating and deleting a file Reading and writing text files Working with directories in PHP Checking for existence of file Determining file size Opening a file for writing, reading, or appending Writing Data to the file Reading characters, Disk Access, I/O, And Mail File upload File download Environment variables E-mail in PHP	**8**

9	**9. Introduction To Database :** Introduction to SQL Connecting to the MYSQL Database creation and selection Database table creation, update table structure insert, update, delete data to a table Fetch data from table, Acquiring the value, Joins, sub query Finding the number of rows Executing multiple queries	**14**
10	**10. Error Handling and PHP Debugging:** Different Types of Error Techniques for finding and fixing bugs in scripts, Exceptional Handling	**4**
11	**11. PHP Config and PHP Security :** PHP Configuration File Popular configuration options Set at runtime to change the behavior of PHP Security issues in web applications Security issues in PHP script Safe Mode on a shared web server	**6**
12	**12. AJAX (Asynchronous JavaScript and XML):** Introduction to AJAX Introduction to XMLHttpRequest Object Method and Properties of XMLHttpRequest Application of AJAX in web application	**8**

REFERENCE BOOKS

1. Beginning PHP7, Apache, Mysql Web Development – Wrox

2. PHP Bible, 2nd Edition: Tim Converse, Joyce Park

3. PHP manual.

MODE OF DELIVERY:

Lectures, Case studies, Practical's, peer discussions and Group work

COURSE ASSESSMENT:

Assessment	Practical	Final Exam	Total
Contribution	40%	60 %	100 %

4. COURSE NAME: (CSE 124) INTRODUCTION TO COMPUTER HARDWARE

LEVEL: YEAR ONE, SEMESTER-II

CREDIT UNITS: 3

CONTACT HOURS: 45

COURSE DESCRIPTION:

Computer hardware is the physical components that a computer system requires to function. It encompasses everything with a circuit board that operates within a PC or laptop; including the motherboard, graphics card, CPU (Central Processing Unit), ventilation fans, webcam, power supply, and so on.

COURSE OBJECTIVES:

This course intends:

1. To enable students understand concepts, techniques, and terminologies about computers.
2. To enable students learn about Design of basic computer.
3. To enable students know registers, various types of registers and interfacing various registers.
4. To enable students learn about the architecture of common bus system.
5. To enable students learn about the different micro-operations used.
6. To enable students learn about Instruction Cycle, Interrupt Cycle.
7. To enable students learn about I/O interface, DMA controller, modes of data transfer and various address modes.
8. To enable students learn how to assemble a PC

LEARNING OUTCOMES:

Upon successful completion of this course students should be able to:

1. Start a small business enterprise by liaising with different stake holders.
2. Effectively manage small business enterprise or working under wage employment.
3. Installation of all the software with cope with different operating system.
4. Develop computer system configuration.
5. Conduct diagnostics - testing and inspection.
6. Have Knowledge of hardware components and latest development in the field.
7. Conduct repair and maintenance of PC's Carry out installation of operating system and applications and have knowledge of Networking and system connectivity.

COURSE OUTLINE:

Unit No	Contents	No Of Lectures
1	**1.Introduction** 1.1. Meaning of Maintenance 1.2. Types of Maintenance 1.3. Hardware and Software Maintenance 1.4. Safety and preventive maintenance procedures 1.5. CRTs and LCD Monitor maintenance 1.6. Safety, health and environmental considerations in computer maintenance	**2**
2	**2.System Troubleshooting** 2.1. Errors and Problem detection techniques 2.2 Computer Error codes and sound Troubleshooting 2.3. Boot/Start-up errors 2.4. Device errors 2.5. Hardware and Software (operating system) errors	**5**
3	**3.System Repair** 3.1. Errors And Problem detection techniques 3.2. Computer system parts/devices 3.3. Operating system: installation and replacement 3.4. Hardware: installation and replacement 3.5. Software: installation and replacement	**3**
4	**4.System Assembly** 4.1. Types and components 4.2. Form factors and dimensions 4.3. Installation and upgrade 4.4. Motherboard interface connectors 4.5. Expansions slots 4.6. Power Supply installation	**2**
5	**5.Secondary Storage Media** 5.1. External storage: installation, configuration, troubleshooting 5.2. Hard disk storage 5.3. Flash storage/removable storage: Formatting memory cards 5.4. Obsolete storage media	**3**

REFERENCE BOOKS:

1. Jean, A. (2016). ComTIA A+ Guide to IT Technical support + lab Manual. 9th edn: Programme Technology Ptr. Mike, M. (2007).
2. Managing And Troubleshooting PCs. 5th edn: McGraw-Hill Education

MODE OF DELIVERY:

Lectures, Case studies, Practical's, peer discussions and Group work

COURSE ASSESSMENT:

Assessment	Practical	Final Exam	Total
Contribution	40%	60 %	100

5. COURSE NAME: (MRG 125) MATHEMATICAL REASONING AND GEOMENTRY (Mathematics)

LEVEL: YEAR ONE, SEMESTER-II

CREDIT UNITS: 3

CONTACT HOURS: 45

COURSE OBJECTIVES:

The course aims to:

1. Introduce students to coordinate geometry, mathematical statements which are relevant to the understanding of object-oriented programming.
2. Provide students with an in-depth knowledge.

LEARNING OUTCOMES:

Upon successful completion of this course students should be able to relate the knowledge acquired to:

i. Develop a solid grasp of fundamental mathematical concepts and logic.
ii. Understand the principles of mathematical reasoning, including induction, deduction, and proof techniques.
iii. Enhance problem-solving skills by applying mathematical reasoning to algorithm design.
iv. Create efficient and effective algorithms based on sound mathematical principles.
v. Analyze the computational complexity of algorithms using mathematical reasoning.
vi. Understand Big O notation and use it to evaluate the performance of code.
vii. Learn methods for formally verifying the correctness of algorithms and programs.
viii. Apply mathematical proofs to ensure that programs meet their specifications.
ix. Use mathematical reasoning to model real-world problems.
x. Translate mathematical models into executable code.
xi. Geometry
xii. Understand fundamental concepts in geometry, including shapes, transformations, and properties of space.

xiii. Learn about coordinate systems and geometric representations.

xiv. Develop algorithms for solving geometric problems such as collision detection, mesh generation, and spatial indexing.

xv. Implement geometric transformations such as translation, rotation, and scaling.

xvi. Study computational geometry techniques for dealing with geometric data.

xvii. Apply algorithms for tasks such as convex hulls, Voronoi diagrams, and Delaunay triangulation.

xviii. Use geometric principles to create and manipulate graphical objects in computer graphics.

xix. Implement algorithms for rendering, shading, and visualization of 3D models.

xx. Simulation and Modeling:

xxi. Apply geometric reasoning to simulate physical systems and environments.

xxii. Develop models for simulations in areas like robotics, physics, and virtual reality.

COURSE OUTLINE

Unit No	**Contents**	**No Of Lectures**
1	1**. Mathematical Reasoning** 1.1 Mathematically acceptable statements. 1.2 Connecting words or phrases—consolidating the understanding of "if and only if (necessary and sufficient) condition", "implies", "and/or", "implied by", "and", "or", "there exists" and their use through a variety of examples related to real life and mathematics. 1.3 Identifying simple statements from compound statements, and writing compound statements using simple statements. 1.4 Validating statements involving connecting words. Difference between contradiction, converse, and contrapositive	**10**
2	**2. Coordinate Geometry** 2.1 Length of a line. 2.2 Midpoint of a line. 2.3 Parallel and perpendicular lines. 2.4 Co-linearity.	**15**

	2.5 Equation of a straight line. 2.6 Shortest distance of a point from a given line. 2.7 The angle between lines. 2.8 Intersection of lines. 2.9 The section formula. 2.10 The area of a triangle whose vertices are given in point form.	
3	**3. Trigonometry** 3.1 Positive and negative angles. 3.2 Measuring angles in radians and degrees, and conversions form one measure to another. 3.3 Finding trigonometric ratios of sin, tan, cosec, sec, and cot using a right-angled triangle, and application to angles of elevation and depressions. 3.4 Solving triangles using the sine and cosine rule. 3.5 Truth of the identity $sin^2x + cos^2x = 1, \forall x$. 3.6 Trigonometric ratios of $30^0, 45^0, and\ 60^0$ and extending the concept to finding the exact value of large angles without using a calculator or tables. 3.7 Expressions of $\sin(x \pm y)\ and\ \cos(x \pm y) in\ terms\ of\ siny, sinx, cosy, and\ cosx$ and their simple applications. 3.8. Derivation of $sin\ 2x, cos2x, tan2x, sin3x, cos3x, and\ tan3x$ using the concept of double angles without any further applications, only derivations.	**1**

MODE OF DELIVERY:

Lectures, practicals, discussions, demonstrations

COURSE ASSESSMENT:

Assessment	Practical	Final Exam	Total
Contribution	40%	60 %	100 %

6. COURSE NAME: (CSE126) REAL LIFE PROJECT II [DYNAMIC WEBSITE]

LEVEL: YEAR ONE, SEMESTER-II

CREDIT UNITS: 3

CONTACT HOURS: 45

COURSE OBJECTIVES:

The objectives of the course are to:

1. To enable students, apply the Fundamentals, building blocks and Advance Concepts of PHP using Web forms in complete details which are essentials for beginners or they can implant.
2. To enable students, develop real life projects using C and C++.
3. To enable students implement customer needs.
4. To teach students how to deploy PHP applications.
5. To teach students how to carry out software tests in PHP using

COURSE DESCRIPTION:

This course provides a systematic and thorough introduction to all aspects of PHP Web forms & MSSQL and the mobile Application development. Projects are an increasingly important aspect of modern business, so we begin with the relation between projects and the strategic goals of the organization. We move on to discuss the technical, cultural, and interpersonal skills necessary to successfully manage projects from start to finish. The course emphasizes that project management is a professional discipline with its own tools, body of knowledge, and skills. Concepts are reinforced by case studies covering a wide variety of project types and industries.

LEARNING OUTCOME:

Upon successful completion of the course, students will be able to:

1. Work as an effective member of a team to implement a software based solution that delivers measurable value to an industry or client.
2. Communicate effectively, orally and in writing, with peers, supervisors and commercial clients/stakeholders.
3. Effectively identify and implement a solution to a complex problem that exists within the domain of ICT.
4. Participate effectively in project and artefact reviews with peers, supervisors and clients/stakeholders.
5. Describe the role and importance of project management, configuration and risk management processes when undertaking a software development project.

6. Participate in a group presentation, including a demonstration, to an audience of peers, clients and supervisors.

COURSE OUTLINE:

Students can develop a project work on any of the following according to their supervisor:
1. Student Management System
2. Attendance Management System
3. Payroll Management System
4. Hospital Management System
5. Food Ordering Management System
6. Courier Management System
7. Online Project approval System PHP
8. Online Test Management Application project
9. Lab Test Management PHP
10. Online bus Ticket Booking project
11. Bank Information System
12. Train Ticket Booking System

OR
Identify suitable system to develop under the project using mobile application system or web development as per the guidelines of the institution and the project guide.

MODE OF DELIVERY

Presentations, Project supervision

MODE OF DELIVERY:

Lectures, Assignment, peer discussions and Group work

7. COURSE NAME: (CIT 120) INDUSTRIAL TRAINING I

LEVEL: YEAR ONE, SEMESTER-II

CREDIT UNITS: 2

END OF THE YEAR ONE FIELD ATTACHMENT.

SEMESTER-III

MODULE NAME: MOBILE APPLICATION TECHNOLOGIES

MODULE CODE: CSE – M III

1. COURSE NAME: (CSE 211) FUNDAMENTALS OF PROGRAMMING IN JAVA

LEVEL: YEAR TWO, SEMESTER-I

CREDIT UNITS: 4

CONTACT HOURS: 45

COURSE DESCRIPTION:

Core Java (developed by Sun Microsystems) is an object-oriented language similar to C++, but simplified to eliminate language features that cause common programming errors. Java source code files (files with a *.java* extension) are compiled into a format called *byte code* (files with a *.class* extension), which can then be executed by a Java interpreter. Compiled Java code can run on most computers because Java interpreters and runtime environments, known as *Java Virtual Machines (VMs)*, exist for most operating systems, including UNIX, the Macintosh OS, and Windows. Byte code can also be converted directly into machine language instructions by a just-in-time compiler (JIT).
Java is a general-purpose programming language with a number of features that make the language well suited for use on the World Wide Web. Small Java applications are called Java applets and can be downloaded from a Web server and run on your computer by a Java-compatible Web browser, such as Netscape Navigator or Microsoft Internet Explorer.

COURSE OBJECTIVES:

By the end of this course, the student should be able to:

This course is an introduction to the Java programming language.
To learn Object Oriented Programming language
To handle abnormal termination of a program using exception handling
To create flat files
To design User Interface using Swing and AWT

Will include an introduction to the concepts of object oriented programming and will show how Java
Supports this programming paradigm. Students will learn about the Java environment and will write

Both applets (programs that execute in a Web browser) and applications (stand-alone program). In addition to learning about basic language statements, students will also learn how Java provides Support for such diverse applications as Web pages, multimedia, educational, etc.

LEARNING OUTCOMES:

On completion of the course, the student should be able to:

1. Mastery of basic programming concepts, like program structure and control, Application Program Interface (API), memory management among others;

2. Demonstrate sufficient understanding of Java as a programming language;

3. Develop simple web and stand-alone applications in Java;

4. Work with an Integrate Development Kit like NETBEANS with minimal difficulties; and Differentiate the various programing languages (i.e., in terms of their strength and weakness).

COURSE OUTLINE:

Unit Number	**Content**	**No. of Lectures**
1	**1. Introduction to Java** 1.1. Features of java 1.2. JDK Environment & tools like(java, javac, appletviewer, javadoc, jdb) 1.3. Installation of Java and IDE	2
2	**2. Object Oriented Programming Concept** 2.1. Overview of Programming 2.2. Paradigm 2.3. Classes 2.4. Abstraction 2.5. Encapsulation 2.6. Inheritance 2.7. Polymorphism 2.8. Difference between C++ and JAVA	3
3	**3. Java Programming Fundamental** 3.1. Structure of java program	4

	3.2. Data types 3.3. Variables 3.4. Operators 3.5. Keywords 3.6. Naming Convention 3.7. Decision Making (if, switch) 3.8. Looping(for, while) 3.9. Type Casting	
4	**4. Classes and Objects** 4.1. Creating Classes and objects 4.2. Memory allocation for objects 4.3. Constructor 4.4. Implementation of Inheritance o Simple o Multilevel o Hierarchical 4.5. Implementation of Polymorphism o Method Overloading o Method Overriding 4.6. Nested and Inner classes	4
5	**5. Arrays String and Vector** 5.1 Arrays o Creating an array 5.2 Types of Array o One Dimensional arrays o Two Dimensional array 5.3. Strings 5.4. String – Arrays , String Methods, String Buffer class, Vectors 5.5. Wrapper classes	4
6	**6. Abstract Class , Interface and Packages** 6.1. Modifiers and Access Control Default, public private protected 6.2. Abstract classes and methods 6.3. Interfaces 6.4. Packages o Packages Concept o Creating user defined packages o Java Built in packages 6.5. Java.lang->math 6.6. Java.util->Random, Date, Hash Table	4

7	**7. Exception Handling** 7.1. Exception types 7.2. Using try catch and Multiple catch 7.3. Nested try 7.4. throw , throws and finally 7.5. Creating User defined Exceptions	4
8	**8. File Handling** 8.1. Byte Stream 8.2. character stream 8.3. file IO Basics 8.4. File Operations o Creating file o Reading file(Character, byte) o Writing File (Character, byte)	3
9.	**9. JAVA SWING**	5

References Books

1. The Complete Reference JAVA 2, Herbert Schildt, TMH
2. The Complete Guide to JAVA Database Programming, Matthew Siple, TMH

MODE OF DELIVERY:
Lectures, Practical's and Group work

COURSE ASSESSMENT:

Assessment	Practical	Final Exam	Total
Contribution	40%	60 %	100 %

2. COURSE NAME: (CSE 212) MOBILE APPLICATION DEVELOPMENT

LEVEL: YEAR TWO, SEMESTER-I

CREDIT UNITS: 4

CONTACT HOURS: 45

COURSE DESCRIPTION:

COURSE OBJECTIVES:

The objective of this course is to understand the Android Operating System and develop applications using Google's Android open-source platform.

UNIT NO	DESCRIPTION	NO. OF LECTURES
1	Android – Overview: Introduction, History, Features Categories	2
2	Android Architecture: API Level, Core building blocks	3
3	Android Environment Setup -Android Emulator	2
4	Steps to make Android apps	4
5	Android Application Components	3
6	ANDROID – ACTIVITIES	4
7	How To Publish App Using Google Play Store	2

REFERENCE BOOKS

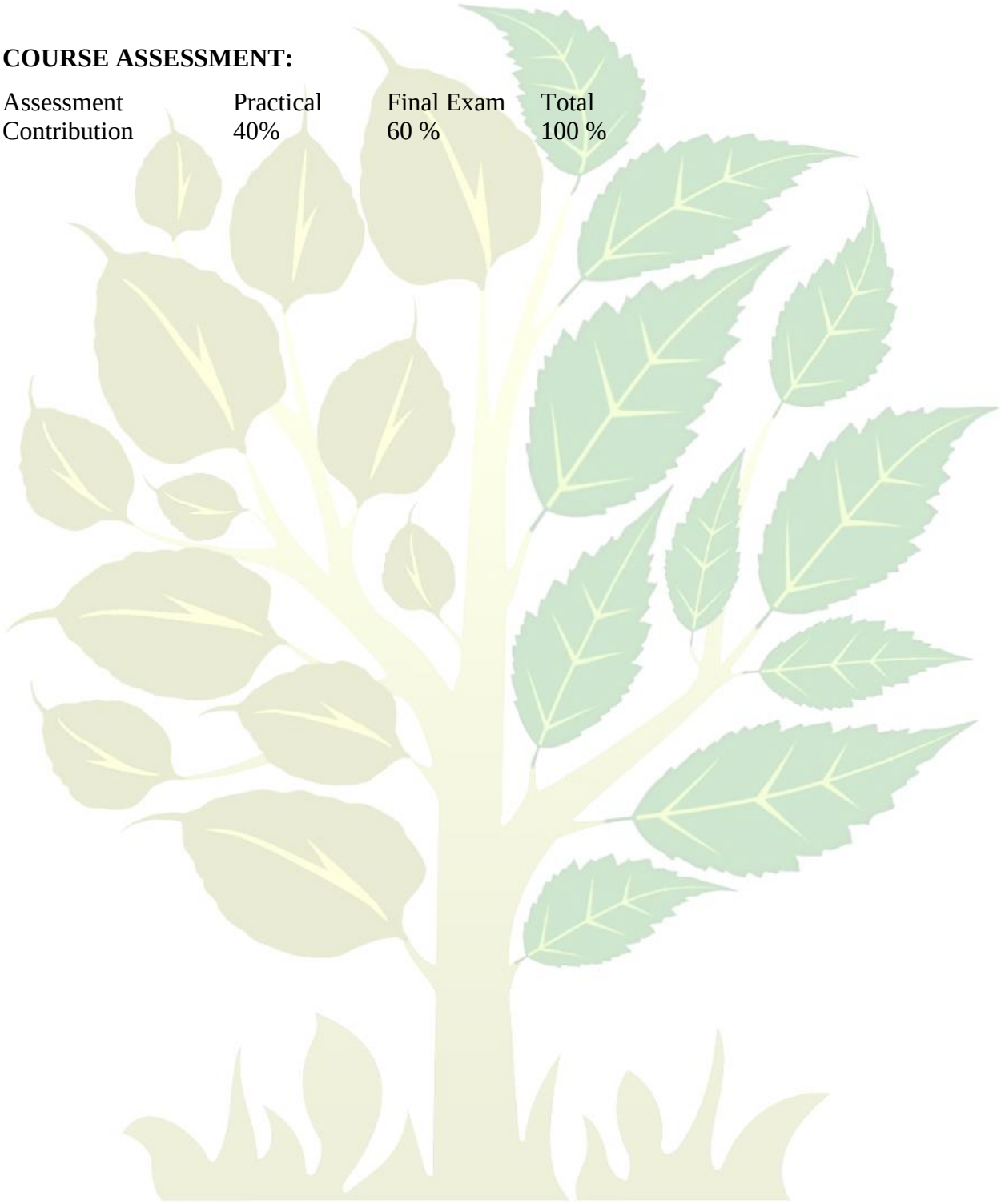
1. Beginning Android4 Application Development, By Wei-Meng Lee WILEY India Edition WROX Publication
2. Professional Android 4 Application Development, By Reto Meier WROX Publication
3. The official site for ***Android developers*** - https://developer.android.com

MODE OF DELIVERY:

Lectures, Case studies, Practical's, peer discussions and Group work

COURSE ASSESSMENT:

Assessment	Practical	Final Exam	Total
Contribution	40%	60 %	100 %



3. COURSE NAME: (CSE 213) BASICS OF NETWORKING

LEVEL: YEAR TWO, SEMESTER-I

CREDIT UNITS: 3

CONTACT HOURS: 45

COURSE DESCRIPTION:

An interconnection of multiple devices, also known as hosts, which are connected using multiple paths for the purpose of sending/receiving data or media. Computer networks can also include multiple devices/mediums which helps in the communication between two different devices, these are known as Network devices and include things such as routers, switches, hubs and bridges.

COURSE OBJECTIVES:

By the end of this course, the student should be able to:

1. To train students how to connect local area networks (LAN), wide area networks (WAN) and wireless version of both types.
2. To train students how to connect hardware devices and set up internet access.
3. To teach students Network cable connectors
4. To expose students to Transmission Media and Components.
5. To train students how to carry out Internet Connectivity in a network.
6. To train Students how to Troubleshooting a Local Area Network.

LEARNING OUTCOMES:

On completion of the course, the student should be able to:

1. Assist others with network problems.
2. Maintain machines attached to the network.
3. Modify Software based on user needs.
4. Maintain files on a network server
5. Monitor system performance

COURSE OUTLINE:

Unit Number	Content	No. of Lectures
1	**1. INTRODUCTION**: 1.1. Meaning of networks 1.2. Advantages of networking to an organization 1.3. Challenges faced when using computer network 1.4. Strategies to overcome the challenges	10
2	**2.COMPONENTS OF LAN:** 2.1. Node, NIC and Modern 2.2. Access point 2.3. Hub (active and passive) 2.4. Repeaters and Bridge 2.5. Switch and routers	10
3	**3.NETWORK CABLE CONNECTORS:** 3.1. RJ-45 3.2. BNC 3.3. db9 serial pinout 3.4. DB-25(Parallel)	4
4	**4.CLASSIFICATION OF NETWORKS:** 4.1. LAN 4.2. Topologies (Ring, Star, Bus and Hybrid) 4.3. WLAN (WI-FI and Bluetooth) 4.4. WAN	4
5	**5. TRANSMISSION MEDIA AND COMPONENTS** 5.1. Difference between analog and digital signals 5.2. Forms of data transmission 5.3. Layout of various cables and their usages 5.4. Wireless media system 5.5. IP Address classes, ranges and their default subnet masks	5
6	**6. INTERNET CONNECTIVITY** 6.1. Components needed to the internet 6.2. Services offered by internet	5

	6.3. Disadvantages of the internet to an organization 6.4. Creating an email account 6.5. Sending and receiving an Email 6.6. Searching for information on the internet	
7	**7.BASICS OF OPERATING SYSTEM SOFTWARE** 7.1. Client End/Window 7.2. 32 bits and 64 bits OS 7.3. FAT-16/32, NTFs 7.4. Configuration of Disks 7.5. Preparing Partitions and Volumes	4
8	**8.TROUBLESHOOTING A LOCAL AREA NETWORK** 8.1.Steps to diagnose a network problem 8.2.Cause of network failures 8.3.How to prevent causes of network failures 8.4.Troubleshooting basic tools for Window connection 8.5.Unguided/Wireless	4

Reference Books

1. Radia, P. (1999). Interconnections: Bridges, Routers, switches and Internetworking

2. Protocols. 2nd edn. Addison- Wesley. Hansell, C.W., U.S. Patent 2,389,432

MODE OF DELIVERY:

Lectures, Case studies, Practical's, peer discussions and Group work

COURSE ASSESSMENT:

Assessment	Practical	Final Exam	Total
Contribution	40%	60 %	100 %

4. COURSE NAME: (CSE 214) SOFTWARE ENGINEERING PRINCIPLES

LEVEL: YEAR TWO, SEMESTER-I

CREDIT UNITS: 3

CONTACT HOURS: 45

COURSE DESCRIPTION:

This course you will learn software engineering principles that will help you to develop better computer software as a good programmer. In this course, you will be able to systematically Approach the design, development, operation, and maintenance of a software system.

COURSE OBJECTIVES:

1. To equip students with software lifecycle and the development process.
2. To equip students with knowledge on of how software engineering helps to manage software development within a division of labor.
3. To teach students Requirements Gathering and Analysis
4. To teach students how to Analyzing the System performance.
5. To teach students how to Design the System
6. To teach students Software Configuration Management

LEARNING OUTCOMES:

On completion of this course, the student will be able to:
1. Understand the key concerns that are common to all software development processes.
2. be able to select appropriate process models, approaches and techniques to manage a given software development process.
3. be able to elicit requirements for a software product and translate these into a documented design.
4. be able to identify dependability and security issues that affect a given software product.
5. Understand the role that testing and reuse play in the implementation phase and how these activities relate to the wider software process.

COURSE OUTLINE:

Unit No	**Contents**	**No Of Lectures**
1	**1. Introduction** 1.1. Describe software problems	**3**

	1.2. Describe the characteristics of the software as a process 1.3. Describe the characteristics of the software as a product 1.4. Describe the need for the Software Engineering discipline 1.5. Define Software engineering goals 1.6. Discuss the role of a software Engineer in its current context	
2	**2. Software Development Process** 2.1. Describe the importance of a phased approach to software development 2.2. List the various phases involved in software development process 2.3. Describe how the various phases are organized into process	**5**
3	**3. Software Development Life Cycle** 3.1. compare how software development companies organize their development process 3.2. Explain the essentials of any process models 3.3. Discuss the advantages and disadvantages of various process model 3.4. Describe the criteria for choosing the appropriate process model 3.5. Describe the process model for the web and the process models 3.6. Compare how software development companies organize their development process	**3**
4	**4. Requirement Gathering and Analysis** 4.1. Describe the process of system analysis 4.2. Explain how a FeasibilityStudy is undertaken 4.3. Describe the scope and activities of the Requirements phase of SDLC 4.4. Explain the process ofnRequirement Gathering	**3**
5	**5. Analyzing the system** 5.1. Describe the System elements 5.2. Discuss Data Modeling 5.3. Explain the concepts of cardinality and modality 5.4. Examine Entity relationship diagram 5.5. Discuss function modelling using DFDs	**3**
6	**6.Designing the system** 6.1. Describe the design process 6.2. Describe the principles of design 6.3. Describe the software design concepts 6.4. Describe the design tools	**3**

7	**7.Software configuration Management** 7.1. Describe baseline concepts 7.2. Describe various configuration items 7.3. Describe the configuration management process	**3**
8	**8.Software Maintenance** 8.1. Describe different types of maintenance 8.2. Describe various maintenance items 8.3. Explain the maintenance metrices 8.4. Describe the working of a typical maintenance organization 8.5. Describe various factors affecting maintenance 8.6. Describe different types of maintenance 8.7. Describe various maintenance issues	**3**

REFERENCE BOOKS

1. Software Engineering for beginners: You're Guide to Creating Software Products – Nico Loubser
2. Software Engineering: A practical Approach – GAOPANDE LAXMIDHAR

MODE OF DELIVERY:
Lectures, Case studies, Practical's, peer discussions and Group work

COURSE ASSESSMENT:

Assessment	Practical	Final Exam	Total
Contribution	40%	60 %	100 %

5. COURSE NAME: (ISP 215) INTRODUCTION TO STATISTICS AND PROBABILITY (Mathematics)

LEVEL: YEAR TWO, SEMESTER-I

CREDIT UNITS: 3

CONTACT HOURS: 4

LEARNING OUTCOMES

By the end of this course, the student should be able to relate the knowledge acquired to:

i. Integrate statistical analysis into mobile applications to enhance functionality and user experience.
ii. Use statistics to provide insights, recommendations, and personalized experiences within mobile apps.
iii. Probability
iv. Incorporate probability-based features in mobile applications, such as recommendation systems, game mechanics, and predictive analytics.
v. Use probability to enhance app functionalities like error handling, user behavior prediction, and personalized content delivery.
vi. Understand the principles of mobile app design, including user interface (UI) and user experience (UX) design.
vii. Learn programming languages and frameworks for mobile app development, such as Java/Kotlin for Android or Swift for iOS.
viii. Develop skills in handling data within mobile apps, including data storage, retrieval, and synchronization.
ix. Implement databases and APIs to manage app data efficiently.
x. Learn to integrate statistical and probabilistic models into mobile applications.
xi. Use libraries and frameworks to facilitate the implementation of these models.
xii. Performance Optimization:
xiii. Optimize mobile apps to handle statistical computations efficiently.
xiv. Ensure that statistical and probabilistic features do not compromise app performance or user experience.
xv. Apply statistical and probability concepts to real-world mobile app scenarios, such as health monitoring apps, financial apps, and social media apps.
xvi. Develop mobile apps that leverage statistical analysis and probability to provide valuable insights and functionalities to users.
xvii. Combined Learning Outcomes
xviii. Interdisciplinary Application:
xix. Apply statistical and probability concepts to solve problems in mobile application development.
xx. Use these concepts to enhance app functionalities, provide data-driven insights, and improve user experience.
xxi. Enhanced Analytical and Problem-Solving Skills:

xxii. Develop strong analytical skills by integrating statistical analysis and probability theory with mobile app development.

xxiii. Solve complex problems by applying a combination of statistical, probabilistic, and programming knowledge.

xxiv. Use statistical and probabilistic methods to make data-driven decisions in the context of mobile app development.

xxv. Implement features that rely on data analysis and probability to offer personalized and intelligent app experiences.

COURSE OUTLINE

Unit Number	**Content**	**No. of Lectures**
1	**1. Statics** 1.1.Measure of central tendencies: mean, mode, and median as applied to both grouped and ungrouped data. 1.2.Measures of dispersion: variance, standard deviation, quartiles, deciles and percentiles as applied to both grouped and ungrouped data. 1.3.Histograms and Ogive curves.	**15**
2	**2. Probability** 2.1.Random experiments; out comes, sample space (set representation). 2.2.Events; occurrence of events, NOT, AND, and OR events, exhaustive events, mutually exclusive events. 2.3.Probability tree diagrams 2.4.Axiomatic (set theory) probability. 2.5.Conditional probability. 2.6.Total probability, Bayes' theorem. 2.7.Discreet and continuous random variables and its probability distribution, mean, and variance of a random variable. 2.8.Repeated independent (Bernoulli) trials and Binomial distribution.	**15**

MODE OF DELIVERY:

Lectures, Assignment, peer discussions and Group work

COURSE ASSESSMENT:

Assessment	Assignment	Test	Final Exam	Total
Contribution	10%	30%	60 %	100%

6. COURSE NAME: (CSE216): REAL LIFE PROJECT III [MOBILE APPLICATION]

LEVEL: YEAR TWO, SEMESTER-I

CREDIT UNITS: 3

CONTACT HOURS: 45

COURSE OUTLINE:

Identify suitable system to develop under the project using mobile application system or web development as per the guidelines of the institution and the project guide.

MODE OF DELIVERY

Presentations, Project supervision

SEMESTER-IV

MODULE NAME: FOUNDATIONS OF DATA SCIENCE AND ARTIFICIAL INTELLIGENCE

MODULE CODE: CSE – M IV

1. COURSE NAME: (GAI 221) GENERATIVE AI TOOLS AND TECHNIQUES

LEVEL: YEAR TWO, SEMESTER-II

CREDIT UNITS: 4

CONTACT HOURS: 45

COURSE OBJECTIVE:

1. Introduction of Generative AI.
2. To understand use of generative AI.
3. To get hands on knowledge of Generative AI Tools & Techniques

COURSE OUTLINE:

Unit Number	Experiments	No. of Lectures
1	**1. Introduction to Generative AI** 1.1. What is Generative AI? 1.2. How Does Generative AI Works? 1.3. Use Cases for Generative AI. 1.4. How Can Businesses Use Generative AI Tools? 1.5. What are the benefits of generative AI? 1.6.What are the limitations of generative AI?	**5**
2	**2. Practical's on Text generation tools include GPT, Jasper, AI-Writer and Lex.**	**5**
3	**3. Practical's on Image generation- using tools (Any one) include Dall-E 2, Midjourney and Stable Diffusion, starryai, Craiyon)**	**5**
4	**4. Practical's on Video Generation- using tools (Any one) Synthesia, Lumen5, Flexclip, Elai, Veed.io.**	**5**

5	**5. Practical's on Music generation- - using tools (Any one) include Amper, Dadabots and MuseNet.**	**5**
6	**6. Practical's on Code generation -using tools (Any one) include CodeStarter, Codex, GitHub Copilot and Tabnine.**	**5**
7	**7. Practical's on Voice synthesis - using tools (Any one) include Descript, Listnr and Podcast.ai.**	**5**
8	**8. Practical's on AI Avatar Generators- using tools (Any one) include-Hippo Video, Picsart, HeyGen, Fotor.**	**5**
9	**9. Practical's on Design Generators - using tools (Any one) include-Khroma, Colormind, Designs.ai, Fronty AI**	**5**

MODE OF DELIVERY:

Lectures, Assignment, peer discussions and Group work

COURSE ASSESSMENT:

Assessment	Assignment	Test	Final Exam	Total
Contribution	10%	30%	60 %	100

2. COURSE NAME: (CSE 222) INTRODUCTION TO PYTHON

LEVEL: YEAR TWO, SEMESTER-II

CREDIT UNITS: 4

CONTACT HOURS: 45

COURSE DESCRIPTION:

This course provides an introduction to programming in Python language. Students are introduced to core programming concepts like data structures, conditionals, loops, variables, and functions. This course includes an overview of the various tools available for writing and running Python and gets students coding quickly. It also provides hands-on coding exercises using commonly used data structures, writing custom functions, and reading and writing to files. This course may be more robust than some other introductory python courses, as it delves deeper into certain essential programming topics.

COURSE OBJECTIVES:

The course aims to provide students with understanding and skills on:
1. To teach leaners why Python is a useful language for developers.
2. To train leaners how to design and program Python applications.
3. To teach students how to use lists, tuples, and dictionaries in Python programs.
4. To train leaners how to identify Python object types.
5. To teach students how to use indexing and slicing to access data in Python programs.
6. To define the structure and components of a Python program.
7. To train leaners how to design object-oriented programs with Python classes.
8. To teach students how to use class inheritance in Python for reusability.

LEARNING OUTCOMES:
A student who successfully completes the course will have the ability to:
1. Identify core aspects of programming and features of the Python language.
2. Understand and apply core programming concepts like data structures, conditionals, loops, variables, and functions.
3. Use different tools for writing and running Python code.
4. Design and write fully-functional Python programs using commonly used data structures, custom functions, and reading and writing to files.

COURSE OUTLINE:

UNIT NO.	DESCRIPTION	NO. OF LECTURES

1	**1. Comments, Variables and Numbers** 1.1. Single and Multiline Comments 1.2. Variables 1. Assign Multiple Variables 2. Output Variables 3. Global Variables 1.3. Data Types 1.4. Numbers 1.5. Operators	5
2	**2. Strings And Lists** 2.1. What is string 2.2. Different string methods 2.3. Lists 2.4. Different operations on Lists	3
3	**3.Tuples, Sets, Dictionary** 3.1. Tuples 3.2. Sets 3.3. Dictionaries	3
4	**4. LOOPS, FUNCTIONS** 4.1. if-else 4.2. while lops 4.3. for loops 4.4. Functions	6
5	**5.Arrays, Lambda** 5.1. lambda 5.2. Arrays 5.3. Arrays Methods	10
6	**6. Object – Oriented Programming (OOP)** 6.1. Define a class 6.2. Instantiate an objects 6.3. Inherit from other classes 6.4. Iterators	3

	6.5. Polymorphism 6.6. Scope	
7	**7. Data** 7.1. Modules and packages 7.2. Working with modules 7.3. Working with packages	4
8	**8.File Input And Output** 8.1. Files and the file system 8.2. Working with file paths in python 8.3. Common file system operations 8.4. Reading and writing files	6
9.	**9. Installing Packages With PIP** 9.1. Installing third-party packages with pip 9.2. pitfall of third-party packages 9.3. summery and additional resources	4
10.	**10.Thinker** 10.1. Python GUI library	5

REFERENCE BOOKS

1. Python for Everybody: Exploring Data Using Python 3 – Charles Severance
2. Programming Python – Mark Lutz
3. Python projects for Beginners – Conner P. Milliken
4. Python Programming or the Absolute beginner – Book by Michael Dawson

MODE OF DELIVERY:

Lectures, Case studies, Practical's, peer discussions and Group work

COURSE ASSESSMENT:

Assessment	Practical	Final Exam	Total
Contribution	40%	60 %	100 %

3. COURSE NAME: (CSE 223) WEB DEVELOPING USING PHP-II

LEVEL: YEAR TWO, SEMESTER-II

CREDIT UNITS: 4

CONTACT HOURS: 45

COURSE DESCRIPTION:

PHP – a server-side scripting language created by Rasmus Lerdorf in 1995 – is installed on more than 200 million of web sites and more than 2 million of web servers. As it is open source by nature and an efficient programming language, it is today's one of the widely used languages for web development. Thus, there is a huge demand for PHP programmers.

COURSE OBJECTIVES:

1. To Design dynamic and interactive Web pages.
2. To Develop Dynamic Web applications.
3. To use database on website

LEARNING OUTCOMES:

1. Learn Core-PHP, Server-Side Scripting Language
2. Learn PHP-Database handling.

COURSE OUTLINE:

Unit No	Contents	No Of Lectures
1	**Working With the File System:** Creating and deleting a file Reading and writing text files Working with directories in PHP Checking for existence of file Determining file size Opening a file for writing, reading, or appending Writing Data to the file Reading characters, Disk Access, I/O, And Mail File upload File download Environment variables E-mail in PHP	8

2	**Introduction To Database** Introduction to SQL Connecting to the MYSQL Database creation and selection Database table creation, update table structure insert, update, delete data to a table Fetch data from table, Acquiring the value, Joins, sub query Finding the number of rows Executing multiple queries	**14**
3	**Error Handling and PHP Debugging** Different Types of Error Techniques for finding and fixing bugs in scripts, Exceptional Handling	**4**
4	**PHP Config and PHP Security** PHP Configuration File Popular configuration options Set at runtime to change the behavior of PHP Security issues in web applications Security issues in PHP script Safe Mode on a shared web server	**6**
5	**AJAX (Asynchronous JavaScript and XML)** Introduction to AJAX Introduction to XMLHttpRequest Object Method and Properties of XMLHttpRequest Application of AJAX in web application	**8**
6	**Advanced Concepts** PHP with XML and AJAX Web Services with PHP, Introduction to Zend Frame work , Introduction to Joomla (CMS) , PHP Best Practices, Code management, Code style, Documentation	**7**

REFERENCE BOOKS

1 Beginning PHP7, Apache, Mysql Web Development – Wrox

2 PHP Bible, 2nd Edition: Tim Converse, Joyce Park

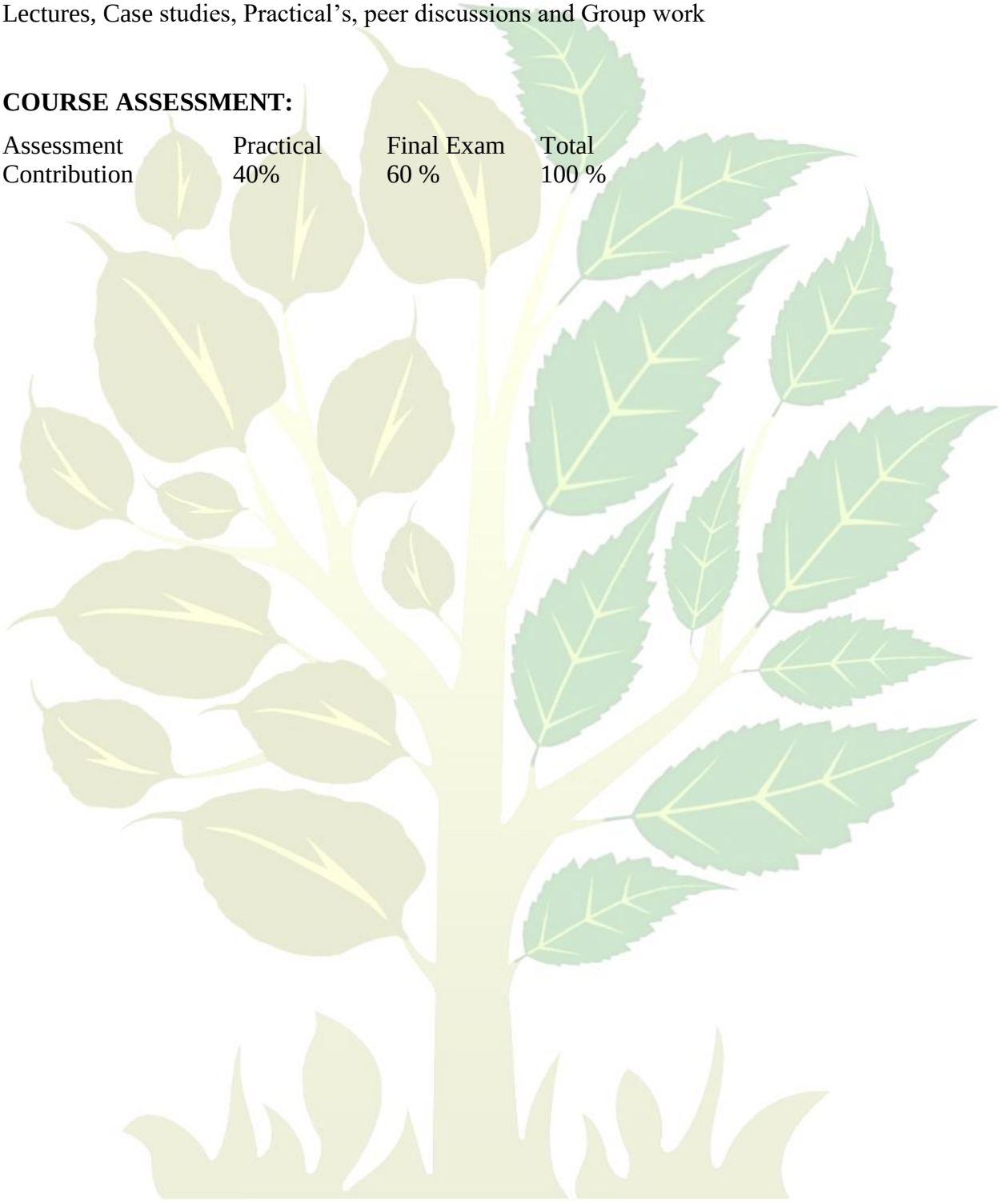
3 PHP manual.

MODE OF DELIVERY:

Lectures, Case studies, Practical's, peer discussions and Group work

COURSE ASSESSMENT:

Assessment	Practical	Final Exam	Total
Contribution	40%	60 %	100 %



4. COURSE NAME: (LAC 224) LINEAR ALGEBRA AND CALCULUS (MATHEMATICS)

LEVEL: YEAR TWO, SEMESTER-II

CREDIT UNITS: 3

CONTACT HOURS: 45

LEARNING OUTCOMES

By the end of this course, the student should be able to use the acquired knowledge to:

i. Develop and implement efficient algorithms for data analysis and AI model training using mathematical concepts.
ii. Analyze and interpret complex datasets by applying linear algebra and calculus principles.
iii. Optimize machine learning and AI models for better performance and accuracy using advanced mathematical techniques.
iv. Implement advanced machine learning techniques that rely on linear algebra and calculus, such as support vector machines and deep learning
v. Use mathematical reasoning to solve scientific computing problems in data science and AI domains.

COURSE OUTLINE:

UNIT NO.	DESCRIPTION	NO. OF LECTURES
1	**1. Differential Calculus** 1.1.Computing limits of polynomials, rational functions, and trigonometric functions. 1.2.Finding derivatives of simple polynomials from first principles. 1.3.The product rule (only in polynomials). 1.4.The quotient rule (only in polynomials). 1.5.The chain rule (only in polynomials). 1.6.Differentiation of parametric equations (only in polynomials). 1.7.Second derivatives (in polynomials). 1.8. Curve sketching of quadratic functions. 1.9. Displacement, velocity, and acceleration problems. 1.10. Tangents and normal.	10

2	**2.Integral Calculus** 2.1 Integration of polynomials only, for both definite and indefinite integrals. 2.2 Applications of integration in motion involving Displacement, Velocity, and Acceleration. 2.3 Applications of integration involving Work Done by a Force.	10
3.	**3. Matrices** 3.1 Concept, notation, order, equality, types of matrices. 3.2 Transpose of a matrix. 3.3 Multiplication of matrices with a scalar 3.4 Addition and multiplication of matrices. 3.5 Determinant of 3x3 matrix 3.6 Application of determinants in finding the area of a triangle. 3.7 Solving a system of linear equations in two or three ONLY by the method of determinants. 3.8 Rotations and transformations. 3.9 Simple application of matrices.	10
4.	**4. Vectors** 4.1 Types of vectors (equal, unit, zero, parallel, and collinear vectors), position vector of a point, negative of a vector, components of a vector. 4.2 Addition of vectors and multiplication of a vector by a scalar. 4.3 The unit vector. 4.4 The ratio theorem. 4.5 The dot product and its applications. 4.6 The cross product	15

MODE OF DELIVERY:

Lectures, Case studies, peer discussions and Group work

COURSE ASSESSMENT:

Assessment	Assignment	Practical	Final Exam	Total
Contribution	10%	30%	60 %	100

5. COURSE NAME: (PES 225) PRINCIPLE OF ENTREPRENEURSHIP SKILL

LEVEL: YEAR ONE, SEMESTER-II

CREDIT UNITS: 3

CONTACT HOURS: 45

COURSE DESCRIPTION:

This module will equip the learner with creative and innovative skills and ability to look out for opportunities by manipulating these natural and man-made resources into business. It is intended to make a learner develop a positive career attitude towards entrepreneurship as a means of making a living. It covers creativity and innovation, scanning the environment for business opportunities, planning a business, managing a business and entrepreneurial ethics.

COURSE OBJECTIVES:

By the end of this course, the student should be able to:

1. To equip students with concepts of entrepreneur and entrepreneurship.
2. To teach students the characteristics/qualities of an entrepreneur.
3. To teach the student the types of entrepreneurs.
4. To teach the students the roles of an entrepreneur.
5. To teach students the entrepreneurship processes.
6. To teach the students the barriers to entrepreneurship development.

LEARNING OUTCOMES:

On completion of the course, the student should be able to:

1. Learns brainstorm the difference between entrepreneur and entrepreneurship.
2. Discuss qualities of a good entrepreneur.
3. How to create a business.
4. Barriers to entrepreneurship development.

COURSE OUTLINE:

Unit Number	**Content**	**No. of Lectures**

1	**1. INTRODUCTION TO ENTREPRENEURSHIP** 1.1.Definition 1.2.Concepts in entrepreneurship 1.3.Reasons /importance of studying entrepreneurship	5
2	**2. The Entrepreneur** 2.1.Types of Entrepreneurs 2.2.Qualities /characteristics of an entrepreneur 2.3.Roles of an entrepreneur in an enterprise	5
3	**3. Entrepreneurial Culture** 3.1.Concept of entrepreneurial culture 3.2.Habits that promote entrepreneurial development 3.3.Factors inhibiting entrepreneurial development 3.4.Ways of managing factors that inhibit development of Entrepreneurial culture	5
4	**4. Creativity And Innovation** 4.1.Meaning of creativity and innovation 4.2.Process of creativity and innovation 4.3.Importance of creativity and innovation 4.4.Barriers to creativity and innovation 4.5.Managing creativity and innovation	5
5	**5. Starting a Small Business** 5.1.Procedure of starting a small enterprise 5.2.Factors to consider when starting a small enterprise 5.3.Legal forms of business ownership 5.4.Challenges faced when starting a small enterprise 5.5.Business life cycle 5.6.Regulations affecting business 5.7.Business support services available for a small business	5

6	**6. Introduction To Business Plan** 6.1. Definition of business plan 6.2. Components of a business plan 6.3.Uses of a business plan 6.4.Preparation of a business plan	5
7	**7. ENTERPRENUERSHIP ETHICS** 7.1.Definition 7.2.Examples of entrepreneurship ethics 7.3.Importance of entrepreneurship ethics 7.4.Ethical challenges facing entrepreneurs 7.5.Solutions to the ethical challenges facing entrepreneurs	

References Books

1. Twenty-five years of Spiral of Silence: A Conceptual Review and Empirical Outlook. International journal of public Opinion Research. 12. Pp. 3-28.
Doi:10.1093/ijpor/132.1.3. Bowman E.(july 2011) .
2. Entrepreneur Training Manual. 3rd edn: Certified Entrepreneur Workbook. Guanzi Institute press. Bruder, Jesica (September 2013)

MODE OF DELIVERY:

Lectures, Practical's and Group work

COURSE ASSESSMENT:

Assessment	Practical	Final Exam	Total
Contribution	40%	60 %	100 %

6. COURSE NAME: (CSE 226): REAL LIFE PROJECT IV [WEB DEVELOPMENT IN PHP USING PYTHON / AI TOOLS]

LEVEL: YEAR TWO, SEMESTER-II

CREDIT UNITS: 3

CONTACT HOURS: 45

COURSE OBJECTIVES:

The objectives of the course are to:
1. To train students how to develop Student Management System.
2. To train students how to develop Attendance Management System.
3. To train students how to develop Payroll Management System.
4. To train students how to develop Hospital Management System.
5. To train students how to develop Food Ordering Management System Python & PostgreSQL Project.
6. To train students how to develop Courier Management System Python Projec
7. To train students how to develop Online Test Management Application Project.
8. To train students how to develop Lab Test Management PHP / Python Project.
9. To train students how to develop Online Bus Ticket Booking Project.
10. To train students how to develop Budget Monitoring System Python Project.
11. To train students how to develop Bank Information System Python Project.
12. To train students how to develop Train Ticket Booking System Python Project.
13. To train students how to develop Regional Transport Office System Python Project.

COURSE DESCRIPTION:

This course provides a systematic and thorough introduction to all aspects of Python & **PostgreSQL**. Projects are an increasingly important aspect of modern business, so we begin with the relation between projects and the strategic goals of the organization. We move on to discuss the technical, cultural, and interpersonal skills necessary to successfully manage projects from start to finish. The course emphasizes that project management is a professional discipline with its own tools, body of knowledge, and skills. Concepts are reinforced by case studies covering a wide variety of project types and industries.

LEARNING OUTCOME:

Upon successful completion of the course, students will be able to:
1. Work as an effective member of a team to implement a software- based solution that delivers measurable value to an industry or client.

2. Communicate effectively, orally and in writing, with peers, supervisors and commercial clients/stakeholders
3. Effectively identify and implement a solution to a complex problem that exists within the domain of ICT.
4. Participate effectively in project and artefact reviews with peers, supervisions and clients/stakeholders.
5. Describe the role and importance of project management, configuration and risk management processes when undertaking a software development project.
6. Participate in a group presentation, including a demonstration, to an audience of peers, clients and supervisors.

COURSE OUTLINE:

Students can develop a project work on any of the following according to their supervisor

1. Student management System
2. Attendance Management System
3. Hospital Management System
4. Payroll Management System
5. Food Ordering Management System Python and PostgreSQL
6. Courier Management System Python Project
7. Online Project Approval System python project
8. Online test Management System PHP/Python project
9. Lab Test Management System PHP/Python project
10. Online Bus Ticket Booking project
11. Bank information System Python project
12. Train ticket Booking system python project

MODE OF DELIVERY

Presentations and Project supervision

COURSE ASSESSMENT

1. Project Proposal	20%
2. Project Demonstration	50%
3. Project Report	30%

7. COURSE NAME: (CIT 220) INDUSTRIAL TRAINING II

LEVEL: YEAR TWO, SEMESTER-II

CREDIT UNITS: 2

END OF THE YEAR ONE FIELD ATTACHMENT.