Kathmandu University School of Education
Department of STEAM Education

Bachelor of Technical Education (B Tech Ed) Program
Information Technology (IT)

In Association with School of Engineering, KU

Course Curriculum

Semester III

2022

Table of Contents

Overview of the B Tech Ed Program

Technical and Vocational Education and Training (TVET) has been one of the prominent sectors globally for a long. Dewey (1916) saw TVET as a tool for education reform in modernizing society. This has been equally vibrant in the present context as scholars (e.g., Maclean & Wilson, 2009) argue that TVET is an education and training that prepares an individual for gainful employment. TVET programs equip young people with the skills, knowledge, and competencies required to enter a particular type of professional career (OECD 2017). Hence, the role of TVET in preparing skilled human resources and enabling them to transition into the career path for gainful employment in specific and the country's economic development in a broad is inevitable. The formal TVET began in Nepal only around mid of the 20th century. Nevertheless, occupational skills have been transformed from generation to generation for a long (Ministry of Education [MoE], 2012). The establishment of the Council for Technical Education and Vocational Training (CTEVT), the governing body of TVET in 1989, was a substantive effort toward its development and expansion (CTEVT, 2019). Additionally, at present, the Centre for Education and Human Resource Development (CEHRD) offering technical diplomas in 284 technical stream schools throughout the country (GoN, 2018) and universities (such as Kathmandu University, Manmohan Technical University, etc.) run bachelor and master in TVET and other 11 federal ministries offer TVET related formal and non-formal programs (MoEST, 2019). The CTEVT, which is in charge of formulating TVET-relevant policies, developing standards for programs and curriculums, coordination, accreditation, monitoring, and supervision (CTEVT, 2019) largely shares the TVET related activities. There are 1131 CTEVT affiliated and constituent institutions with roughly the average capacity of 70,000 per annual operate formal TVET programs (CTEVT, 2020). Further expansion of TVET institutions in the country is yet to be expected soon since the government has a provision of establishing at least one technical school in all local levels.

Bachelor of Technical Education is a program to develop instructors and teachers in the field of Technical and Vocational Education and Training (TVET) with a view to fulfil the gap of acute shortage of instructors and TEVT professionals in the country which aspires to promote technical and vocational education two-third of the educational system and processes. The program has envisaged that a mix of two or more of the work-based learning, activity-based instruction, on the job training, project-based learning and problem-based learning be used to deliver the program. Because of the ‘practical focus’ of the program, the assessment has been proposed to address the needs of experiential learning.

The 136-credit program has been structured four types of the courses, such as Educational Core (18), Vocational Specialization (80), Vocational Pedagogy (20), and Educational Research and Internship (18) with a view to inculcate both pedagogical and vocational skills. The vocational specializations have been offered in a manner that they can be utilize hands-on activities as well. The program begins with the foundational courses in general education, vocational pedagogy and the area of specialization. In the later part of the program, OJT/WBL and forms of experiential learning shall be introduced. The culmination of the program will be internships in their specializations and a research report that addresses the problem of vocational education.

B Tech Ed in Information Technology

Bachelor of Technical Education is a 4-year (8 semesters) program to develop instructors, teachers, professionals in the field of Technical and Vocational Education and Training (TVET) with a view to fulfil the gap of acute shortage of instructors and TEVT professionals in the country which aspires to promote technical and vocational education two-third of the educational system and processes. The program has envisaged that a mix of two or more of the work-based learning, activity-based instruction, on the job training, project-based learning and problem-based learning be used to deliver the program. Because of the 'practical focus' of the program, the assessment has been proposed to address the needs of experiential learning. The B Tech Ed in IT program aims at providing technical, vocational, and educational exposures for students who can become IT professionals, IT Instructors, TVET educators, TVET curriculum developers, etc. The occupational/specialization courses emphasize the development of core IT skills in the field of computer science and technology innovation. The B Tech Ed in IT graduates will be eligible for doing master's program in IT, TVET, Education, Social Sciences, etc.

Major Objectives of the B Tech Ed in IT Program

- Demonstrate Level 6 competencies envisaged by the National Vocational Qualification Framework (NVQF) of Nepal,
- Exhibit comprehensive, meaningful, and coherent knowledge and skills in the field of technology and computer science
- Apply occupational specialization knowledge and skills in solving problems occurred in their contexts of work,
- Apply vocational pedagogical knowledge and skills in technical and vocational instruction of Information Technology,
- Show scholarly literacy, communication, quantitative reasoning, critical thinking, learning skills needed for advanced and work-based learning,
- Exemplify a deep and principled understanding of the technical and vocational learning processes and the role of the instructors in facilitating these processes in the students,
- Apply a wide range of teaching process skills including curriculum development, lesson planning, materials development, assessment, and pedagogical approaches, and
- Enable students to become resilient and lifelong learners by harnessing internal and external assets as TVET professionals.

Career Prospects

- TVET Instructors and Educators
- TVET Curriculum Developers
- IT Professionals (Software and Apps developer, programmer, Network Engineer, Web Developer, IT Admin, IT Engineer, etc.)
- Eligible for doing master's program in IT, TVET, Education, Social Sciences, etc.

Entry Requirements

The minimum prerequisite for this program is the completion of Level 5 education (as per National Vocational Qualification Framework) in relevant vocational specializations. The details are-

- Minimum 1.6 CGPA in 10+2
  Or
- A minimum 40% marks in 10+2, PCL or equivalent in any disciplines

Courses Offered in the Semester III

Subject	Credit Hour		
	Theory	**Practical**	**Total**
Core Course			
EDUC 200 Educational Psychology	2	1	3
Vocational Pedagogical Course			
VPED 210 Instructional Skills (I)	1	1	2
Occupational Specialization Course			
ITEX 200 Data Structure and algorithm	2	1	3
ITEX 214 IT Project Management	1	1	2
ITEX 201 Computer Networks	2	1	3
ITEX 212 Microprocessor and computer architecture	2	1	3
ITEX 213 Principles of Internet Technologies and Web applications	2	1	3
Total	12	7	19

Core Courses

EDUC 200 Educational Psychology

Course Name: Educational Psychology **Code:** ITEX 200
Credit Hour: 3 (2+1) **Nature:** Theory + Practical

Course Description and Goals

In the context of education, the need for knowledge and skills of educational psychology is crucial. The effectiveness of instructional activities and the learning performance of students depend upon the skills of instructors and curriculum developers in understanding various educational and psychological aspects. Educational psychology provides the platform to connect psychological aspects to education. The educational stakeholders must have knowledge and skills in educational psychology. This course is designed for undergraduate students to develop knowledge and skills in educational psychology, focusing on the usage of it in constructing effective learning environments. The students will get exposure to various aspects of educational psychology ranging from understanding the learners to learning diversity in the classroom. Moreover, the course will engage students actively in the learning process and activities to practically apply and develop a deeper understanding of each concept and content.

Learning Outcomes

The following are the learning outcomes that students will develop:

1. Explore the meaning of educational psychology for education,
2. Develop knowledge and skills of learning the learners by focusing on the various developmental aspects such as physical, brain, language, and cognitive development,
3. Create a deeper understanding of the learning process and learning theories focusing on behaviorism, cognitivism, constructivist, and humanist learning perspectives,
4. Develop knowledge on diversities of learners and learning differences and utilizing the concepts in classroom interactions and instructions,
5. Apply the concept of educational psychology in designing lessons, activities, and materials!

Content with specific objectives

Specific Objectives	Contents
● To define and describe educational psychology ● To describe the nature and principal of learning ● To discuss the and interpret the purpose of educational psychology ● To express the roles of educational psychology	**Module I: Educational Psychology for Learning & Teaching (3 hours)** ● Nature and Principals of Learning ● Purpose of Educational Psychology ● Roles of Educational Psychology
● To describe and analyze the developmental system perspectives based on variable,	**Module II: The Learner (21 hours)** 1. Developmental System Perspectives a. Development as variable b. Development as relational

relational, influenced by context, and influenced by the child) ● To analyze the physical development of person overtime and explain the educational implication of each level ● To describe brain development in different stages (Infancy, Early Childhood, Middle Childhood, and Adolescence) and discuss the roles of education in each developmental stage ● To explain and analyze the development of language in different stages (Infancy, early childhood, middle childhood, and adolescence) ● To describe the roles of adults and peers in language development ● To explain Piaget's theory of cognitive development and stages of cognitive development with educational implications ● To explain and interpret Vygotsky's sociocultural theory with its educational implication ● To discuss and explain the information processing components of cognitive development and Metacognition ● To explain and interpret meanings and roles of social, emotional, and moral development in the case of understanding self, others, and relations with others	c. Development as influenced by context d. Development as influenced by the child 2. Physical Development Overtime a. Development in Infancy b. Development in early childhood c. Development in Middle Childhood d. Development in Adolescence 3. Brain Development a. Brain Development in Infancy b. Development in early childhood c. Development in Middle Childhood d. Development in Adolescence 4. Language Development a. Language Development in Infancy b. Language Development in early childhood c. Language Development in Middle Childhood d. Language Development in Adolescence e. The Roles of adults and Peers in language development 5. Cognitive Development a. Piaget's Theory of Cognitive Development b. Vygotsky's Social Cultural Theory c. Information Processing Components of Cognitive development 6. Social, emotional, and Moral Development a. Understanding Self (Self-development, sense of self, and identity) b. Understanding Others (Moral development) c. Understanding Relations with Others (Development of social competence, relationship with peers, bullying and harassment)
● To interpret the meaning and roles of behaviorism in education ● To describe the educational implication and limitation of behaviorism theory of learning ● To explain cognitive learning theory with the major focus on	**Module III: The Learning Process (9 hours)** 1. Behaviorist View of Learning a. Introduction and Principles of Behaviorist Explanation of Learning b. Educational Implication of Behaviorist Learning 2. Cognitive and Constructivism Views of Learning

Piaget & Vygotsky and explaining their educational implication in teaching and learning ● To explain and interpret the meaning and educational implication of constructivism ● To create the examples of learning environment, role of students, and teachers in behaviorist, cognitivist, constructivist, humanist theories of learning ● To provide examples and explanation of Maslow's hierarchy of needs and Rogers' Non-directive teaching and 'freedom to learn' concepts ● To explain the meanings of humanism and education, and humanism in the classroom with the examples	a. Cognitive Learning Theory (Piaget & Vygotsky) (Meaning, Key Principles, and classroom applications) b. Constructivism (Meaning, Key Principles, and classroom applications) 3. Humanist Views of Learning a. Humanism & Psychology i. Maslow and Hierarchy of Needs ii. Rogers: Non-directive teaching and 'freedom to learn' b. Humanism & Education c. Humanism in the classroom
● To explain the nature of a diverse classroom environment ● To define and explain motivation and engagement in learning with examples ● To describe the meanings and roles of intelligence and creativity in learning ● To describe Gardner's Theory of Multiple Intelligence (MI) including examples and classroom implication ● To discuss the ideas of ability and potential in the context of gifted and talented students ● To describe creativity and teaching creativity with examples ● To explain learner diversity and learning support needs with their emphasis on educational implications	**Module IV: Individual Difference in the Classroom (12 hours)** 1. Motivation and Engagement a. Defining motivation and engagement b. Key concepts in motivation c. Engagement in Learning 2. Intelligence and Creativity a. What is intelligence? b. Gardner's Model of Intelligence (MI Theory) c. Ability and potential (Gifted and talented students) d. Creativity and Learner 3. Learning Support & Inclusive Education a. Learner diversity and schooling b. Learning support needs (Intellectual and cognitive differences, Attention Deficit Hyperactivity Disorder (ADHD), Autism Spectrum Disorders (ASD), Physical Disabilities, Sensory and Speech disabilities, etc.)

● To describe and explain the usages of classroom differentiation and universal design for learning (UDL) to address the diverse needs of students ● To interpret and exemplify the sociocultural factors (Gender; Ethnicity, language, and culture; and socio-economic status)	c. Teaching and Diverse needs (Classroom differentiation & Universal Design for Learning) 4. Sociocultural Factors in Learning Process a. Gender b. Ethnicity, language, and culture c. Socio-economic status

Major Teaching and Learning Strategy

The facilitator(s) can use multiple methods or strategies for teaching and learning listed below. Students must be actively engaged in each and every learning activity. Frequent sharing sessions and reflection activities should be conducted. So, teaching and learning should be based on a constructivist model.

- Lecture Method
- Inductive & Deductive Methods
- Collaborative learning (Peer and group learning)
- Discovery and investigation approach
- Project-based learning
- Inquiry-based approach
- Demonstration approach
- ICT integrated approach

Assessment Plan

d. In-semester: 50%

i. Assignments & Reflective writings,
ii. Collaborative group work (At least 3),
iii. Presentations (At least One individual and one group),
iv. Classroom participation,
v. Learning demonstration activities
vi. **Projects:** At least one of the following projects must be carried out during teaching and learning activities

a) **Module II:** Collect data of child development (Physical, Brain, Language, Social, emotional, and Moral development) by using observation and interview methods; analyze the data by connecting various ideas of child developed you studied and develop a report. Discuss what significant changes you explored.

b) **Module III:** Observe at least one regular class of a schoolteacher (grades 1-12), note the major teaching and learning activities and characteristics, teacher materials, strategies, and evaluate his/her class based on Behaviorist, Cognitivist, Constructivist, and humanist theories of learning. Also, provide feedback to the teacher (or make a list of suggestions/recommendations) to improve the practice.

c) **Module IV:** Do one observation of students of a particular class of any school (or technical school) and study the diversity of students based on engagement, motivation, intelligence, creativity, special issues (ADHD,

ASD, Physical disabilities, etc. if available). Prepare a learning model or teaching and learning activities plan (differentiated instruction or UDL) to address the diverse needs of students.

e. End-semester: 50%

References

Duchesne, S., McMaugh, A., & Mackenzie, E. (2022). *Educational psychology for learning and teaching* (7th ed.). Cengage Learning.

Woolfolk, A. (2021). *Educational psychology* (14th ed.). Pearson.

https://www.verywellmind.com/

Vocational Pedagogical Course

VPED 210 Instructional Skills (I)

Course Name: Instructional Skills (I) **Code:** VPED 210
Credit Hour: 2 (1 TH + 1 PR) **Nature:** Theory + Practical

Course Description and goals

This course is designed to provide students with a firm understanding of instruction and its management system. It expects students to understand instructional supervision model with some practical experiences.

Learning Outcomes

- Understand the foundation for effective instruction
- Be familiar with concept of the instructional management system
- Develop and apply Model of instructional supervision cycle
- Identify appropriate instructional key skills for 21st Century and demonstrate their use in the field
- Understand Occupational Health and Safety (OHS) guidelines in real setting

Content with Specific objectives

Specific Objectives	Contents
➢ To understand the foundation and approach of instruction ➢ To be familiar with the guiding principles and dimensions of effective instruction	**1. Foundation & Approach of Instruction (3 hours)** 1.1 Effective Instruction 1.2 Global Approach of Instruction 1.3 Guiding Principles of Effective Instruction 1.4 Dimensions of Instructions 1.5 Dimensions of Effective Instructions
➢ To understand the concepts and types of instructional management ➢ To understand strategies, techniques and management system of instruction ➢ To develop PDCA cycle for instructional management ➢ To develop and manage classroom culture	**2. Instructional Management System (6 hours)** 2.1 Concepts & Types of Instructional Management 2.2 Instructional Strategies & techniques 2.3 Instructional Management System with PDCA cycle 2.4 Develop and Manage Classroom Culture
➢ To understand and be familiar with instructional key skills	**3. Instructional Key Skills (6 hours)** 3.1 Role Skills 3.2 Personal Skills 3.3 Active Listening Skills 3.4 Research & Evaluation Skills 3.5 Collaboration & Problem-Solving Skills 3.6 Ethics & Accountability 3.7 21st Century Key Skills • *ICT and Digital Skills*

➢ To be familiar with the models of instructional supervision ➢ To able to use clinical supervision during practical experience ➢ To identify and manage classroom stress ➢ To prepare effective assessment and evaluation techniques	**4. Models for Instructional Supervision (10 hours)** 4.1 Models of Instructional Supervision 4.2 Clinical Supervision • *Pre-Observation, Observation, & Post-Observation* 4.3 Classroom Stress Management & Supervision 4.4 Assessment & Evaluation Techniques
➢ To prepare students for the field exposure in the real world setting for the quality instructional skills and for its effective implementation.	**5. Practicum Session (5 hours)** 5.1 Occupational Health and Safety (OHS) 5.2 Observe and prepare case study on Clinical Supervision 5.3 Observe and demonstrate classroom instruction with 21st century skills 5.4 Observe and apply Key Instructional Skills for effective classroom management 5.5 Effective ways to implement classroom culture and Problem-solving skills

Major teaching and learning strategy

- Lecture classes and demonstration, Project Based Learning (PBL), and Group Work.
- Regular assessments (Five numbers)

  Practical (Field Visit): *List of tasks*
- Students will visit technical institutes and observe the teacher's teaching practice, maintain class log book, Lesson (Session) plan, use instructional methodology (Oral, Presentation, Research), Yearly Plan Operation (YPO), techniques and strategies (Use of Clinical Supervision) – **12 hours**

Assessment plan

In semester: 50

- First Internal
- Second Internal
- Project Based Learning - Practical Demonstration
- Field Visit Report

End Semester: 50

- MCQ: 10
- Subjective: 40

Textbooks

- Barry, K., & King, L. (1998). *Beginning teaching and beyond* (3rd ed.). Social Science Press.
- Marsh, C. (2000). *Handbook for beginning teachers.* Addison Wesley Longman.
- Farrant, S. (1980). *Principles and practices of education.* Longmans
- Mc Burney-Fry, G. (1998). *Improving your practicum: A guide to better teaching practice.* Social Science Press Australia.

- Saskatchewan Education. (1991). *Instructional approaches: A framework for professional practice.*

Useful links:

https://thoughtfulclassroom.com/
https://www.britishcouncil.org/school-resources/develop-skills/online-courses/exploring-instructional-leadership

Occupational/Specialization Course

ITEX 200 Data Structure and Algorithm

Course Title: Data Structure and Algorithm **Course Code:** ITEX 200

Credit Hours: 3 (2+1) **Nature of the course:** Theory and Practical

Course Description and Goals

The Data Structures and Algorithms course is designed to provide students with a solid conceptual understanding of various data structures and algorithms, enhance their problem-solving skills, foster algorithmic thinking, and improve their programming abilities. Students will develop a strong foundation in this field by studying topics such as Stack, queue, linked list, trees, graphs, sorting, searching, and hashing algorithms. They will learn to analyze algorithms, evaluate their efficiency, and make informed decisions when selecting and optimizing algorithms for different scenarios. Practical exercises and programming assignments will enable students to efficiently apply data structures and algorithms to solve real-world problems. Ultimately, the course aims to equip students with the necessary knowledge, skills, and mindset to excel in the field of data structures and algorithms, preparing them for further studies and careers in related disciplines.

Learning Outcomes

On completion of this course, the students will be able to:

- Describe and explain basic data structures like Stack, queue, and linked list, including their implementation, operations, and applications.
- Analyze algorithms using time and space complexity, allowing for evaluating and comparing their efficiency and performance.
- Utilize a variety of sorting, searching, and hashing algorithms for solving problems and manipulating data effectively.
- Understand the fundamentals of trees and graphs, including their properties, representations, and algorithms for traversal and manipulation.
- Apply data structures and algorithms to solve real-world problems, demonstrating problem-solving skills and the ability to choose appropriate strategies.

Content with specific objectives

Specific objectives	Contents
• understand the importance of Data Structures • describe the characteristics of an Algorithm • Applying Asymptotic Analysis • Identifying Algorithmic Notations • assess the Relationship Between Data Structures and Algorithms	**Unit 1: Introduction (4 Hours)** 1.1 Definition of Data Structure, Abstract Data Type, Importance of Data Structure i. Definition of data structure and abstract data type (ADT) ii. Importance of data structures in organizing and manipulating data efficiently iii. Examples of real-world applications of data structures 1.2 Introduction to Algorithms, Characteristics of an Algorithm i. Introduction to algorithms and their role in Problem-solving ii. Characteristics of an algorithm:

	Finiteness, Definiteness, Input, Output, Effectiveness 1.3 Asymptotic Analysis and Notations i. Introduction to asymptotic analysis for evaluating algorithm efficiency ii. Best-case, worst-case, and average-case analysis scenarios a) Big O notation: Upper bound notation to describe the worst-case complexity b) Omega notation: Lower bound notation to describe the best-case complexity c) Theta notation: Tight bound notation to describe both upper and lower bounds
• understand the Stack as an Abstract Data Type (ADT) • apply Stack Operations • analyze Stack Behavior • Evaluate Stack Applications • create Stack-Based Solutions	**Unit 2: The Stack (4 Hours)** 2.1 Introduction to Stacks as an Abstract Data Type (ADT): i. Definition of a stack and its characteristics ii. LIFO (Last-In, First-Out) principle of a stack iii. Basic operations supported by a stack: PUSH and POP iv. Understanding the Stack as a linear data structure 2.2 PUSH and POP Operations, Full and Empty Stack: i. Explanation of the PUSH operation to add elements to the Stack. ii. Explanation of the POP operation to remove elements from the Stack. iii. Understanding the concept of a full stack and an empty stack. iv. Illustrating the behavior of the Stack with examples and visualizations. 2.3 Stack Applications: Evaluation of Expressions: i. Introduction to the use of stacks in expression evaluation. ii. Evaluation of infix expressions using the Stack. iii. Conversion of infix expressions to postfix and prefix forms using stacks. iv. Examples and exercises on expression evaluation and conversion. 2.4 Conversion of Expressions: i. A detailed explanation of the conversion process for infix expressions. ii. Algorithmic steps for converting infix expressions to postfix and prefix forms.
• Understand the Queue as an Abstract Data Type (ADT) • To Apply Queue Operations	**Unit 3: Queue (4 Hours)** 3.1 Introduction to Queue:

• To analyze Queue Behavior • Evaluating Queue Applications • To Explore the priority queue	i. Definition of a queue and its characteristics as an Abstract Data Type (ADT). ii. Understanding the First-In, First-Out (FIFO) principle of a queue. iii. Introducing the concept of queue operations and their importance in managing data. 3.2 Queue as an ADT: i. Exploring the properties and behavior of a queue as a linear data structure. ii. Understanding the abstract interface of a queue, including enqueue and dequeue operations. iii. Differentiating a queue from other data structures based on its characteristics. 3.3 Primitive Operations in Queue: i. Explaining the basic operations of a queue: enqueue, dequeue, and peek. ii. Discussing the purpose and functionality of each operation. iii. Demonstrating the behavior of a queue through practical examples. 3.4 Linear and Circular Queue and Their Applications: i. Differentiating between linear and circular queues. ii. Understanding the advantages and disadvantages of each queue implementation. iii. Discussing real-life applications that utilize linear and circular queues. 3.5 Enqueue and Dequeue Operations: i. Detailing the enqueue operation to add elements to a queue. ii. Explaining the dequeue operation to remove elements from a queue. iii. Discussing the concept of front and rear pointers in managing a queue. 3.6 Priority Queue: i. Introducing the concept of a priority queue and its purpose. ii. Exploring different strategies for implementing priority queues.
• understand Static and Dynamic List Structures • apply Array Implementation of Lists • Analyze List-Based Queues	**Unit 4: List and Linked List (7 Hours)** 4.1 Static and Dynamic List Structure: i. Differentiate between static and dynamic list structures. ii. Understand the advantages and limitations of each approach.

• understand Linked Lists as an Abstract Data Type (ADT) • apply Insertion and Deletion Operations on Linked Lists • analyze Insertion and Deletion after and before Nodes • evaluate Linked Stack and Queue Implementations • analyze Circular and Doubly Linked Lists	iii. Discuss scenarios where static or dynamic list structures are suitable. 4.2 Array Implementation of Lists: i. Implement a list using an array data structure. ii. Understand the operations and complexities associated with array-based lists. iii. Discuss the advantages and limitations of array-based list implementations. 4.3 Queues as a List: i. Explore the use of lists as a foundation for implementing queues. ii. Understand how list operations can be used to mimic queue behavior. iii. Discuss the similarities and differences between list-based queues and traditional queues. 4.4 Linked List as an ADT: i. Introduce the concept of a linked list and its characteristics as an ADT. ii. Understand the structure and components of a linked list. iii. Discuss the advantages and limitations of linked lists compared to array-based lists. 4.5 Insertion and Deletion of Node to and from a List: i. Explain the process of inserting a new node into a linked list. ii. Understand the various scenarios and techniques for node deletion from a linked list. iii. Analyze the complexity of insertion and deletion operations in linked lists. 4.6 Insertion and Deletion after and before Nodes: i. Discuss the concept of inserting a new node after or before a specific node in a linked list. ii. Demonstrate the process of node insertion and deletion based on node position. iii. Explore the practical applications and use cases for these operations. 4.7 Linked Stack and Queues: i. Understand the implementation of stacks and queues using linked lists. ii. Discuss the advantages and disadvantages of using linked lists for stack and queue implementations. iii. Analyze the complexities and behaviors of linked list-based stack and queue operations.

	4.8 Circular and Doubly Linked Lists and Their Advantages: i. Introduce circular linked lists and understand their structure and behavior. ii. Discuss the advantages and applications of circular linked lists. iii. Introduce doubly linked lists and compare them to singly linked lists in terms of functionality and advantages.
• Understand Basic Operations in Binary Trees • Apply Tree Search and Insertion/Deletion • Analyze Binary Tree Traversals and Tree Metrics • Evaluate Types of Binary Trees • Understand Balanced Trees and AVL Balancing	**Unit 5: Trees (7 Hours)** 5.1 Basic Operations in Binary Tree: i. Understand the fundamental operations performed on a binary tree, such as creating a tree, accessing nodes, and updating nodes. ii. Discuss the key concepts of parent, left child, and right child nodes in a binary tree. iii. Demonstrate the implementation of basic operations in binary trees. 5.2 Tree Search and Insertion/Deletion: i. Explain the process of searching for a specific value or node in a binary tree. ii. Understand the techniques for inserting and deleting nodes in a binary tree. iii. Analyze the complexities and trade-offs associated with search, insertion, and deletion operations in binary trees. 5.3 Binary Tree Traversals (pre-order, post-order, and in-order), Tree Height, Level, and Depth: i. Explain and demonstrate the three fundamental tree traversal techniques: pre-order, post-order, and in-order. ii. Understand the concept of tree height and how to calculate it. iii. Discuss the notion of tree-level and depth and their significance in tree analysis. 5.4 Strictly Binary Tree, Complete Binary Tree, Almost Complete Binary Tree: i. Define and differentiate between strictly binary trees, complete binary trees, and almost complete binary trees. ii. Understand the properties and characteristics of each type of binary tree. iii. Discuss the advantages and applications of different types of binary trees in specific scenarios. 5.5 Balanced Tree: AVL Balanced Trees, Balancing Algorithm:

	i. Introduce the concept of balanced trees and the importance of maintaining balance in tree structures ii. Explain AVL balanced trees and their self-balancing property iii. Understand the balancing algorithm used in AVL trees to maintain balance during node insertion and deletion.
• understand Internal & External Sort • apply Sorting Algorithms (Insertion Sort, Selection Sort, Bubble Sort) • analyze Recursive Sorting (Quick Sort, Merge Sort, Binary Sort) • Evaluate the Efficiency of Sorting using Big 'O' Notation	**Unit 6: Sorting (7 Hours)** 6.1 Internal & External Sort: i. Understand the concepts of internal and external sorting. ii. Differentiate between sorting data that fits in main memory (internal) and sorting data that requires external storage (external). iii. Discuss the advantages, limitations, and techniques used in internal and external sorting. 6.2 Sorting Algorithms (Insertion Sort, Selection Sort, Bubble Sort): i. Introduce and explain the basic sorting algorithms, including insertion sort, selection sort, and bubble sort. ii. Understand the logic, step-by-step process, and time complexity of each algorithm. iii. Compare and analyze the efficiency, strengths, and weaknesses of these sorting algorithms. 6.3 Recursive Sorting (Quick Sort, Merge Sort, Binary Sort): i. Explore recursive sorting algorithms, such as quick sort; merge sort; and binary sort. ii. Understand the recursive nature and divide-and-conquer approach of these algorithms. iii. Analyze the time complexity and efficiency of recursive sorting algorithms. 6.4 Efficiency of Sorting using Big 'O' Notation: i. Introduce the concept of algorithmic efficiency using Big 'O' notation. ii. Analyze the time complexity of sorting algorithms and their efficiency in different scenarios. iii. Compare and evaluate the performance of sorting algorithms based on their Big 'O' complexities.
• Understand Search Techniques and Essential Concepts • apply specific Search Algorithms (Sequential Search, Binary Search, Tree Search, General Search Tree)	**Unit 7: Searching and Hashing (6 Hours)** 7.1 Search Technique and Essential of Search: i. Understand the importance of search techniques in data structures and algorithms. ii. Explore the essential components and characteristics of search algorithms.

• analyze efficiency and performance • understand Hashing: Hash Functions and Hash Tables • evaluate Collision Resolution Techniques	iii. Discuss the different types of searches and their applications. 7.2 Sequential Search, Binary Search, Tree Search, General Search Tree: i. Explain the sequential search algorithm and its step-by-step process. ii. Introduce the binary search algorithm and its efficient approach for sorted data. iii. Discuss tree search techniques, including traversal and search operations in binary search trees. iv. Explore general search trees, such as AVL trees and B-trees, and their search capabilities. 7.3 Efficiency Comparison of Different Search Techniques: i. Analyze and compare the efficiency of sequential search, binary search, and tree-based search techniques. ii. Evaluate the time complexity and performance of each search technique in different scenarios. iii. Discuss the advantages and limitations of different search algorithms. 7.4 Hashing: Hash Functions and Hash Tables: i. Introduce the concept of hashing and its applications in data structures. ii. Explain hash functions and their role in mapping data to hash table indices. iii. Discuss the implementation and operations of hash tables for efficient data retrieval. 7.5 Collision Resolution Techniques: i. Explore collision resolution methods used in hash tables, such as chaining and open addressing. ii. Compare and evaluate the efficiency and trade-offs of different collision resolution techniques. iii. Understand the importance of handling collisions to maintain hash table performance.
• understand Graph as an Abstract Data Type (ADT) • apply Transitive Closure and Warshall's Algorithm • analyze types of Graphs • analyze Graph Traversal and Spanning Trees • evaluate Shortest Path First Algorithm (Dijkstra's Algorithm)	**Unit Eight: Graph (6 Hours)** 8.1 Graph as an Abstract Data Type (ADT): i. Understand the concept of a graph and its representation as an abstract data type. ii. Explore the basic components of a graph, such as vertices and edges. iii. Discuss the operations and functionalities provided by Graph ADT. 8.2 Transitive Closure: i. Understand the concept of transitive closure in a graph.

• evaluate Greedy Algorithms	ii. Learn about Warshall's algorithm to compute the transitive closure of a graph. iii. Apply the algorithm to determine the reachability between vertices in a graph. 8.3 Types of Graphs: i. Explore different types of graphs, including directed and undirected graphs, weighted and unweighted graphs, and cyclic and acyclic graphs. ii. Understand the characteristics and properties of each type of Graph. 8.4 Graph Traversal and Spanning Trees: i. Discuss graph traversal algorithms, such as depth-first search (DFS) and breadth-first search (BFS). ii. Explore the concept of spanning trees and their applications in graph theory. iii. Introduce algorithms like Kruskal's algorithm and Prim's algorithm for finding minimum spanning trees. 8.5 Shortest Path First Algorithm (Dijkstra's Algorithm): i. Understand the concept of the shortest path problem in graphs. ii. Learn about Dijkstra's algorithm for finding the shortest path in weighted graphs. iii. Apply the algorithm to find the shortest path between vertices in a graph. 8.6 Greedy Algorithms: i. Introduce the concept of greedy algorithms in the context of graph problems. ii. Explore applications of greedy algorithms in graph theory, such as finding minimum spanning trees and shortest paths. iii. Discuss the characteristics, advantages, and limitations of greedy algorithms.

Major Teaching and Learning Strategy

Depending on the topic, the standard instructional techniques include in-class lectures, small-group discussions, case studies, guest lectures, lab work, project work, assignments (both theoretical and practical), and exams. According to the demands of the subjects, the teaching staff will select the appropriate pedagogy. The facilitator is in charge of creating the instructional materials needed for the teaching-learning process; multimedia use is advised because visuals and audio will encourage student participation more effectively. At the very least, the facilitator should use a standard programming language to demonstrate how to utilize a data structure.

Assessment Plan

a. In-semester (50 marks)

i. Internal – I

ii. Internal – II
iii. Assignment
iv. Practical Exam/ Viva

b. End-semester (50 marks)

i. MCQ – 10
ii. Subjective – 40

Textbooks and References

1. Ellis, H., Sartaj S., & Dinesh, P. M. (2007). *Fundamentals of data structures in C++*. Silicon Press.

Reference Book:

1. Y. Langsam, M.J. Augenstein and A. M, Tanenbaum, "Data Structures using C and C++", PHI
2. G. W. Rowe, "Introduction to Data Structures and Algorithms with C and C++", PHI.

List of Laboratory Exercises (But not limited to):

1. Stack Operations
    - Implement the stack data structure using an array or linked list.
    - Perform basic stack operations such as push, pop, peek, and display.
    - Create applications that utilize stack operations, such as infix to postfix conversion, parentheses balancing, and evaluating postfix expressions.
2. Queue Operations
    - Implement the queue data structure using an array or linked list.
    - Perform basic queue operations such as enqueue, dequeue, peek, and display.
    - Create applications that utilize queue operations, such as simulation of real-world scenarios like a ticket booking system or a process scheduling algorithm.
3. Linked List Manipulation
    - Implement the linked list data structure.
    - Perform operations like insertion, deletion, and traversal on the linked list.
    - Implement advanced concepts like reversing a linked list or merging two sorted linked lists.
4. Tree Data Structure
    - Implement the binary tree data structure.
    - Perform basic tree operations such as insertion, deletion, and traversal (pre-order, in-order, post-order).
    - Implement additional tree-related algorithms like finding the height of a tree, checking if a tree is balanced, or constructing a binary search tree.
5. Sorting Algorithms
    - Implement various sorting algorithms such as bubble sort, insertion sort, selection sort, merge sort, and quicksort.
    - Compare and analyze the efficiency of different sorting algorithms using time complexity analysis and empirical measurements.

- Implement sorting algorithms for different data types and evaluate their performance.

6. Graph Algorithms
    - Implement graph data structures and operations such as adding vertices and edges.
    - Implement graph traversal algorithms like depth-first search (DFS) and breadth-first search (BFS).
    - Implement graph algorithms like finding a minimum spanning tree using Kruskal's or Prim's algorithm or finding the shortest path using Dijkstra's algorithm.
7. Hashing and Hash Tables
    - Implement hash functions and hash tables using arrays or linked lists.
    - Perform hash table operations like insertion, retrieval, and deletion.
    - Implement collision resolution techniques such as chaining or open addressing.
8. Problem-Solving with Data Structures and Algorithms
    - Solve programming problems that require the application of various data structures and algorithms.
    - Implement efficient solutions for searching, sorting, graph traversal, or dynamic programming tasks.
    - Analyze the time and space complexity of the implemented solutions and optimize them if necessary.

Note: The individual instructor has the flexibility to choose and conduct laboratory sessions on specific topics within the course "Data Structures and Algorithms." This flexibility allows the instructor to tailor the laboratory work to the specific needs and interests of the students. The instructor can select topics that align with the course objectives and cover areas of particular importance or relevance to the student's field of study. By exercising this flexibility, the instructor can create a dynamic and engaging learning environment, fostering hands-on experience and practical application of data structures and algorithms concepts. This approach also encourages creativity and exploration, enabling students to delve deeper into specific areas of interest and comprehensively understand the subject matter.

ITEX 201 Computer Networks

Course: Computer Networks **Course Code:** ITEX 201
Credit Hours: 3 (2 TH + 1 PR) **Nature:** Theory and Practical

Course Description and Goals

The course provides an introduction to computer networks and covers the concepts and principles of networking technologies, including OSI and TCP/IP reference models. Students will learn about the protocols and technologies used in networking, such as *Ethernet, wireless networking, routing,* and *switching*. This course also covers network security, firewalls, intrusion detection systems, and cryptography.

Learning Outcomes

On completion of this course, the students will be able to:

- Describe the basic principles and concepts of computer networking.
- Analyze and interpret OSI and TCP/IP reference models.
- Interpret the strengths and weaknesses of different network devices and infrastructure components such as switches, routers, hubs, and repeaters.
- Develop a practical understanding of networking concepts such as Ethernet, wireless networking, routing, and switching.
- Analyze and evaluate computer network performance and management, including network monitoring, bandwidth management, and QoS.
- Describe network security tools and technologies.

Content with specific objectives

Specific objectives	Contents
• To define and describe on computer network and its applications. • To discuss on network architectures, topologies, data standards. • To describe and differentiate between TCP/IP and OSI reference model and its layers. • To analyze and interpret data communication and its components.	**Unit One - Introduction to Computer Networks (10 hours)** • Computer Network, • Network topologies and architectures, • OSI reference model, • TCP/IP reference model, • network structure, • network architecture, • network standards, • data communication and its components
• define and describe Ethernet and WIFI. • explore different routing protocols used in networking. • define and differentiate between TCP/IP and UDP. • explain DNS and its application.	**Unit Two: Network Protocols and Technologies (8 hours)** • Ethernet and Wi-Fi, • Routing protocols, • TCP/IP and UDP, • Domain Name System (DNS), • IP addressing

• Define IP addressing tools and techniques.	
• define switches and routers, hubs, and repeaters applications in networking. • describe bridges and gateways in network. • Describe network cables and connectors.	**Unit Three: Network Devices and Infrastructure (6 hours)** • Switches and Routers, • Hubs and Repeaters, • Bridges and Gateways, • Network cables and connectors
• explore different types of media used in networking. • discuss various transmission modes, and media signals. • explore network layer tasks, and it's use in networking. • define and describe the internet model.	**Unit Four: Media and Mode (6 hours)** • Media mode, • Media speed, • Transmission mode, • Media signals, • Network layer tasks, • Internet model
• define and describe wireless networking technology and its applications. • discuss various wireless connection technologies such as Wi-Fi, Bluetooth and infrared technology.	**Unit Five: Wireless Networking (5 hour)** • Wireless LANs, • Wi-Fi standards and security, • Wireless mesh networks, • Bluetooth and infrared
• explore network security threats and their vulnerabilities. • define and describe firewalls and intrusion detection systems. • define and describe cryptography and various encryption techniques. • explore VPN and its importance. • discuss network monitoring and troubleshooting techniques and their importance in network security. • explore maintaining Quality of Service in networking applications. • discuss management of bandwidth and speed in the network.	**Unit Six: Network Security (10 hour)** • Network security threats and vulnerabilities, • Firewalls and intrusion detection systems, • Cryptography and encryption, • Virtual private networks (VPNs), • Network monitoring and troubleshooting, • Quality of Service (QoS), • Bandwidth management, • Network design

• To design and implement network design for network security.	

Major Teaching and Learning Strategy

- ***Lectures:*** This will be an executive method where the instructor will provide information on the syllabus's contents.
- ***Presentations:*** Students will present on several topics for the presentation. This will help to understand the student's skills in several concepts of computer networks.
- ***Field visits at ISP and Telecom company:*** Students will be taken to several institutes that have implemented networking. This will help students understand the network design implemented in several places for information communication.
- ***Case studies:*** Both instructors and students will be involved in going through several case studies to understand the implementation of networking. This will be an interactive session, as the understanding will be shared in the regular classroom. So, both instructors are students expected to be active and conduct interactive sessions. Instructors are highly expected to involve students in sharing their knowledge and understanding actively.
- ***Lab works:*** This will be a hands-on practice to understand the concepts discussed in the classroom.

Assessment Plan

In semester (50):

Written exam, Objectives, lab report, assignment, presentation, demonstration, field report, case studies, viva, attendance.

End Semester (50):

Objectives, Subjective

Practical (Some examples):

- To set up a small Local Area Network (LAN) using routers, switches, and Ethernet cables and configuring IP addresses and subnet masks.
- To configure wireless network devices, such as access points and routers, and secure them using WPA2 or other encryption methods.
- To set up and configure a simple network application, such as a file-sharing system or a web server, and test its connectivity.
- Write a Program using C, to find the Even Parity bits/Odd parity bits in a message and append the redundant bit to the original message.
- LAN Router Configuration using Packet Tracer.
- Create VLAN on the switch using Packet Tracer.
- Create 2 VLANS on a switch and configure the router to enable communication between the 2 VLANS.

Textbooks and Reference Materials

- Tanenbaum, A. S. (2011). *Computer networks* (5th ed.). Pearson Education India.
- Tanenbaum, A. S., Feamster, N., & Wetherall, D. (2020). *Computer networks* (6th ed.). Pearson Global Edition.

ITEX 212 Microprocessor and Computer Architecture

Course Title: Microprocessor and Computer Architecture **Code:** ITEX 212
Credit Hours: 3 (2 TH + 1 PR) **Nature:** Theory +Practical

Course Description and Goals

This course provides a comprehensive study of microprocessors and computer architecture. It covers the fundamental concepts and principles underlying the design and operation of microprocessor-based systems. Students will explore topics such as the organization of microprocessors, memory hierarchy, input/output organization, and control unit operations. Through hands-on laboratory exercises, they will gain practical experience in programming microprocessors and implementing computer systems. The course aims to develop students' understanding of microprocessors' internal structure and functioning, enhance their programming skills, and equip them with the knowledge to design and configure computer systems. By the end of the course, students will be able to recall the history and significance of microprocessors, analyze microprocessor-based systems, program using assembly language, and demonstrate knowledge of memory organization, cache memory principles, and input/output organization.

Learning Outcomes

On completion of this course, the students will be able to:

- Recall the history and importance of microprocessors.
- Analyze and explain the organization of microprocessor-based systems.
- Apply register transfer language and micro-operations to describe microprocessor operations.
- Program the 8086-microprocessor using assembly language.
- Understand the structure and function of the control unit and perform arithmetic operations.
- Demonstrate knowledge of memory organization, cache memory principles, and input/output organization.

Content with specific objectives

Specific objectives	Contents
• Recall the key features of microprocessors throughout history and describe the basic block diagram of a computer system. • Understand the organization and architecture of microprocessor-based systems, including the CPU, memory, and interfaces. • Explain the concept of buses and their functions in computer systems. • Understand the stored program concept and its significance in computer architecture and apply the knowledge to explain the processing	**Unit One: Introduction (5 Hours)** 1.1 Introduction and History of Microprocessors (**30** minutes) i. Overview of microprocessors and their historical development. ii. Introduction to key milestones and advancements in microprocessor technology. 1.2 Basic Block Diagram of a Computer (**30** minutes) i. Understanding the components and their interconnections in a computer system. ii. Overview of the CPU, memory, and I/O devices. 1.3 Organization of Microprocessor-Based System (**1** hour)

of instructions and data in a stored program architecture. • Recall different integration technologies (SSI, MSI, LSI, VLSI) used in computer systems and describe their impact on computer architecture. • Apply the knowledge of register transfer language (RTL) to describe data transfer between registers and memory.	i. Detailed study of the organization and architecture of microprocessor-based systems. ii. Exploring the CPU, memory, and various interfaces. 1.4 Bus Organization **(1** hour) i. Understanding the concept of buses and their role in computer systems. ii. Different types of buses (data bus, address bus, control bus) and their functions. 1.5 Stored Program Concept and Von Neumann Architecture (**1** hour) i. Introduction to the stored program concept and its significance. ii. Exploring the principles of Von Neumann architecture. • Processing of a Stored Program (**30** minutes) i. Understanding the execution cycle and processing of instructions and data in stored program architecture. ii. Overview of the fetch-decode-execute cycle and the role of the CPU. 1.7 SSI, MSI, LSI, and VLSI Technology (**30** minutes) i. Introduction to different integration technologies (SSI, MSI, LSI, VLSI). ii. Exploring the trends and advancements in integrated circuit technology. 1.8 Register Transfer Language and Micro-operations **(30 minutes**) i. Introduction to register transfer language (RTL) and micro-operations. ii. Understanding the transfer of data between registers and memory.
• Understand the internal architecture and features of the 8086 microprocessor and its components. • Explain assembly language syntax, including comments, identifiers, instructions, and directives. • Demonstrate the process of assembling, linking, and executing assembly language programs.	**Unit Two: Programming with 8086 Microprocessor (11 Hours)** 2.1 Internal Architecture and Features of 8086 (4 hours) i. Understand the internal architecture of the 8086 microprocessors. ii. Study the components of the Bus Interface Unit (BIU) and Execution Unit (EU). iii. Learn about the operations performed by the EU and BIU. iv. Understand segment and offset addressing and the addressing modes of the 8086.

• Implement various programs in the 8086, including arithmetic, logical, and string operations. • Apply conditions, loops, and array processing techniques in assembly language programming. • Read and display ASCII, decimal, binary, and hexadecimal numbers using assembly language. • Understand the pin configuration, bus structure, and timing of read and write operations in the 8086 microprocessors.	2.2 Assembly Language Programming (6 hours) i. Compare high-level and low-level programming languages. ii. Study the syntax of assembly language, including comments, reserved words, identifiers, statements, directives, operators, and instructions. iii. Learn the process of assembling, linking, and executing assembly language programs. iv. Understand the concepts of one-pass and two-pass assemblers. v. Explore keyboard and video services in assembly language. vi. Implement various programs in the 8086, including arithmetic, logical, and string input/output operations. vii. Learn about conditions and loops in assembly language. viii. Explore array and string processing techniques. ix. Read and display ASCII and decimal numbers in assembly language. x. Display numbers in binary and hexadecimal formats. 2.3 Configuration and Bus Structure (1 hour) i. Understand the pin configuration of the 8086 microprocessor. ii. Study the bus structure, including synchronous and asynchronous buses. iii. Learn about the timing of read and write operations on the bus in the 8086.
• understand the architecture and components of the CPU. • explain the function and purpose of the ALU. • apply arithmetic and logical operations using the ALU. • understand the concept and purpose of stack organization. • perform stack operations and manage the stack effectively. • describe the organization and components of the processor. • analyze the interaction between different parts of the processor.	**Unit Three: Central Processing Unit (7 Hours)** 3.1 CPU Structure and Function (1 hour) i. Understand the overall structure and function of a Central Processing Unit (CPU). ii. Explore the major components and their roles within the CPU. 3.2 Arithmetic and Logic Unit (ALU) (1 hour) i. Study the design and operation of the Arithmetic and Logic Unit (ALU) within the CPU. ii. Understand how the ALU performs arithmetic and logical operations. 3.3 Stack Organization (1 hour) i. Learn about the stack organization within the CPU.

• understand the organization and types of registers. • utilize registers for efficient data storage and manipulation. • Compare and contrast RISC and CISC architectures. • evaluate the advantages and characteristics of each architecture type.	ii. Understand the stack pointer and its role in managing the stack. 3.4 Processor Organization (1 hour) i. Explore the organization and components of the processor. ii. Understand the interaction between different parts of the processor. 3.5 Register Organization (1 hour) i. Study the organization and types of registers within the CPU. ii. Understand the purpose and use of different registers. 3.6 Instruction Formats and Addressing Modes (1.5 hours) i. Understand different instruction formats used in CPU architectures. ii. Explore various addressing modes and their impact on instruction execution. 3.7 Data Transfer and Manipulation (1 hour) i. Learn about data transfer and manipulation operations performed by the CPU. ii. Understand how data is moved between registers, memory, and peripherals. 3.8 RISC and CISC Architectures (0.5 hours) i. Compare and contrast Reduced Instruction Set Computer (RISC) and Complex Instruction Set Computer (CISC) architectures. ii. Understand the characteristics and advantages of each architecture type.
• understand the concept and function of Control Memory. • analyze addressing sequencing and computer configuration in the control unit. • understand the microinstruction format and interpret symbolic microinstructions. • Understand the operation of the control unit and its role in instruction execution and design a control unit based on given specifications.	**Unit Four: Control Unit (6 Hours.)** i. Control Memory - 0.5 hours ii. Addressing sequencing - 0.5 hours iii. Computer configuration - 0.5 hours iv. Microinstruction Format - 0.5 hours v. Symbolic Microinstructions - 1 hour vi. Symbolic Micro program - 1 hour vii. Control Unit Operation - 1 hour viii. Design of control unit - 1 hour

• comprehend the algorithm for performing addition and Subtraction using signed magnitude representation. • comprehend the algorithm for performing addition and Subtraction using signed 2's complement representation. • analyze and apply the array multiplier algorithm for performing multiplication. • apply Booth's algorithm for efficient multiplication of signed numbers. • understand the multiplication algorithm for signed magnitude representation. • analyze and apply the restoring division algorithm for performing division.	**Unit Five: Computer Arithmetic (6 Hours.)** 5.1 Addition and Subtraction (2 hours) i. Algorithm for Signed Magnitude ii. Algorithm for Signed 2's Complement 5.2 Multiplication Algorithm (2 hours) i. Array Multiplier ii. Booth's algorithm iii. Signed Magnitude 5.3 Division Algorithm (2 hours) i. Restoring Division
• Understand the types and functions of peripheral devices. • Comprehend the purpose and operation of I/O modules. • Comprehend the role and features of input-output interfaces. • Understand the different modes of data transfer in input and output operations. • apply the concepts of programmed I/O to perform input and output operations. • apply the principles of interrupt-driven I/O to handle input and output operations efficiently. • apply the concept of direct memory access to transfer data between peripheral devices and memory without CPU intervention. • understand the purpose and functions of a data communication processor.	**Unit Six: Input and Output Organization (5 Hours)** i. Peripheral devices (0.5 hours) ii. I/O modules (0.5 hours) iii. Input-output interface (0.5 hours) iv. Modes of transfer (0.5 hours) v. Programmed I/O (1 hour) vi. Interrupt-driven I/O (1 hour) vii. Direct Memory access (1 hour) viii. Data Communication processor (0.5 hours)

• understand memory systems: • comprehend the characteristics and hierarchy of memory systems. • differentiate between internal and external memory. • explore cache memory: • understand the principles and elements of cache design. • apply cache size, mapping function, replacement algorithm, and write policy concepts.	**Unit Seven: Memory Organization (5 Hours)** 7.1 Memory (2 hours) i. Microcomputer Memory ii. Characteristics of memory systems iii. The Memory Hierarchy iv. Internal and External memory 7.2 Cache Memory (3 hours) i. Principles ii. Elements of Cache Design iii. Cache size iv. Mapping function v. Replacement algorithm vi. Write policy vii. Number of caches

Laboratory Works (But not limited to)

- **Lab Work:** Familiarize students with microprocessor architecture, components, and basic operations. Recommended Software: GNU toolchain, Logisim.
- **Assembly Language Programming:** Lab Work: Develop proficiency in writing and executing assembly language programs. Recommended Software: GNU toolchain, QEMU.
- **Microprocessor Interfacing:** Lab Work: Understand the interaction between microprocessors and peripheral devices. Recommended Software: GNU toolchain, Logisim, QEMU.
- **Control Unit Design:** Lab Work: Design and simulate control units for microprocessors. Recommended Software: Logisim, QEMU.
- **Computer Arithmetic:** Lab Work: Perform arithmetic and logical operations using microprocessors. Recommended Software: GNU toolchain, QEMU.
- **Input and Output Organization:** Lab Work: Interface microprocessors with various input and output devices. Recommended Software: GNU toolchain, QEMU.
- **Memory Organization:** Lab Work: Explore memory systems and implement memory-related operations. Recommended Software: GNU toolchain, Logisim, QEMU.
- **System Integration and Testing:** Lab Work: Integrate different components, test the complete system, and troubleshoot issues. Recommended Software: GNU toolchain, QEMU, OpenOCD.

Hardware-based Laboratory

The Arduino Uno is a versatile microcontroller board that can greatly enhance our course syllabus on computer system architecture. With its wide range of capabilities and ease of use, the Arduino Uno serves as an excellent tool for practical demonstrations, hands-on experiments, and project-based learning.

By utilizing the Arduino Uno in lab sessions, we can provide students with a tangible platform to explore microprocessor architecture, digital and analog signals, and interfacing with various components. They can better understand the theoretical concepts by directly

implementing them on the Arduino Uno and observing the results in real time. The Arduino Uno also offers an opportunity for students to engage in creative exploration and prototyping. They can design and build small-scale projects that showcase different aspects of computer systems and microprocessor operations. This hands-on approach fosters creativity, problem-solving skills, and a deeper connection to the subject matter.

Furthermore, the Arduino Uno aligns with real-world applications and trends in the field. It is widely used in the making community and industry for prototyping and developing embedded systems and IoT devices. Incorporating the Arduino Uno into our syllabus ensures that our students gain practical skills and familiarity with a widely adopted platform, enhancing their future career prospects. In addition, the Arduino Uno facilitates collaborative learning and project-based assignments. Students can work in teams to design and implement projects integrating multiple concepts from our syllabus. This promotes teamwork, communication, and the development of practical project management skills.

Overall, the Arduino Uno empowers students to bridge the gap between theory and practice in computer system architecture. Its practicality, versatility, and real-world relevance make it an invaluable resource for enhancing our course, providing students with a hands-on, engaging, and comprehensive learning experience.

Note: The course instructor has the flexibility to design the laboratory works based on the hardware and software available in their local environment. The specific equipment, tools, and software resources may vary depending on the institution and its resources. The instructor can tailor the Laboratory works to align with the available hardware platforms, development boards, microprocessors, and software tools. This approach allows for a practical and hands-on learning experience relevant to the course's specific context. By leveraging the local resources, the instructor can provide students with opportunities to gain practical skills and knowledge applicable to their future endeavors in computer system architecture.

Major Teaching and Learning Strategy

There are two groups of teaching strategies used in this course. The first group comprises of general teaching strategies that can be used with most units. The second group comprises of specialized teaching methods that can be used with particular units. Students will receive reading materials for each course, and all chapters use lectures, group discussions, multimedia projector use, and brainstorming for general teaching strategy. On the other hand, specialized teaching strategy incorporates practical demonstrations as a key teaching strategy during the teaching and learning process. The specific teaching method for this course will be a demonstration of practical works.

Assessment Plan

1. **In-semester (50 marks)**
    - **a.** Internal – I
    - **b.** Internal – II
    - **c.** Assignment
    - **d.** Practical Exam/Viva/ Design of a hypothetical computer
2. **End-semester (50 marks)**
    - a. MCQ – 10
    - b. Subjective – 40

Text and Reference Books

1. "Computer System Architecture (2007)" by Morris Mano - This book provides comprehensive coverage of computer system architecture, including topics such as microprocessors, memory systems, I/O organization, and control unit design.
2. "Computer Organization and Design (2012)" by David A. Patterson and John L. Hennessy - Widely used in computer science and engineering courses, this book covers computer organization and design principles, including topics such as instruction set architecture, data path and control unit design, memory hierarchy, and I/O systems.
3. "Microprocessor Architecture, Programming, and Applications with the 8085" by Ramesh S. Gaonkar - This book focuses on microprocessor architecture, programming, and applications, specifically centered around the Intel 8085 processor. It covers topics such as instruction sets, addressing modes, memory interfacing, I/O programming, and interrupt handling.

4. "Structured Computer Organization" by Andrew S. Tanenbaum - Offers a structured approach to computer organization and architecture, this book covers topics such as digital logic design, instruction set architecture, processor organization, memory hierarchy, and I/O systems.

ITEX 213 Principles of Internet Technologies and Web Applications

Course Title: Principles of Internet Technologies and Web Applications **Course Code:** ITEX 213
Credit Hours: 3 (2 TH + 1 PR) **Nature:** Theory +Practical

Course Description and Goals

This course is designed to provide students with a comprehensive understanding of the principles of internet technologies and web applications. Students will learn about the evolution of the internet, protocols and standards used throughout the internet, a variety of internet and WWW applications and related technologies. This course will also cover the opportunities and threats created by interconnecting computers via the internet.

Learning Outcomes

On completion of this course, the students will be able to:

- Describe how the internet and web technologies have evolved over time and the impact they have on daily lives.
- Define and describe different applications and technologies used in web development and creating interactive and dynamic websites.
- Explore different types of security risks and vulnerabilities associated with web development and best practices for securing web applications.
- Describe basics of server-side programming, including the different programming languages used to build web applications.
- Analyze different opportunities and challenges associated with internet interconnection, including privacy and data protection issues, technical challenges, policy and governance issues.

Content with specific objectives

Specific objectives	Contents
• To define and describe the history of the internet and web technologies. • To describe client-server model and it's used in web development. • To describe common web development tools and technologies. • To explain ethical and social responsibility considerations in web development.	**Unit 1: Introduction to Internet Technologies and Web Applications (5 hrs.)** • History of internet, • overview of web technologies, • client-server model, • web development tools and technologies, • Ethics and social responsibility in web development

• To discuss key internet protocols and standards. • To explain TCP/IP and OSI reference models. • To describe HTTP and HTTPS protocols and their applications in web development. • To explain on DNS system and domain name resolution. • To explore email protocols.	**Unit 2: Internet Protocols and Standards (6 hrs.)** • Overview of Internet protocols and standards, • TCP/IP and OSI reference models, • HTTP and HTTPS protocols, • DNS and domain name resolution, • Email protocols
• To define and describe key components of web development such as HTML, CSS, jQuery and JavaScript. • To explore responsive web design principles. • To explore on accessibility and usability guidelines in web development.	**Unit 3: Web Development Fundamentals (4 hrs.)** • HTML and CSS basics, • Introduction to JavaScript and jQuery, • Responsive web design principles, • Accessibility and usability guidelines
• To explore server-side programming languages. • To define and explore PHP and Node.js and their applications in web development. • To explore on building and deploying web applications	**Unit 4: Server-Side Programming (8 hrs.)** • Server-side programming languages, PHP, Node.js, • Building and deploying web applications
• To explore web security threats and vulnerabilities. • To explain best practices for securing web applications and the use of SSL and HTTPS. • To analyze and evaluate security risks in web development.	**Unit 5: Web Security (5 hrs.)** • Web security threats, • web security vulnerabilities, • best practices for securing web applications, • SSL and HTTPS
• To describe emerging web technologies, web APIs, web services, microservices, and progressive web apps.	**Unit 6: Emerging Web Technologies (6 hrs.)** • Web APIs, • web services and microservices, • progressive web apps, • future of web development

• To explore the potential of emerging technologies to shape the future of web development.	
• To describe the Internet and WWW applications. • To explore application areas on electronic commerce, online payment systems, social networking, cloud computing. • To express the concept of mobile computing and IoT.	**Unit 7: Internet and WWW Applications (5 hrs.)** • Internet and WWW applications, • eCommerce and online payment systems, • social networking and online communities, • cloud computing and big data, • Mobile computing and the Internet of Things (IoT)
• To explore benefits and risks associated with interconnecting computers via the internet. • To discuss different types of cybersecurity threats and counter measures associated with internet interconnection. • To explain the privacy and data protection issues associated with internet interconnection. • To express on the key policy and governance issues associated with internet interconnection.	**Unit 8: Opportunities and Threats of Internet Interconnection (6 hrs.)** • Benefits and risks of Internet interconnection, • Cybersecurity threats and counter measures, • Privacy and data protection, • Internet governance and policy issues

Major Teaching and Learning Strategy

- *Lectures:* The course will include lectures that provide an overview of the key concepts and principles of internet technologies and web applications, as well as their practical applications.
- *Hands-on exercises:* Students will engage in hands-on exercises, including building and deploying web applications, in order to apply the concepts learned in lectures.
- *Group projects:* Students will work in groups to design and develop web applications, allowing them to apply their knowledge in a collaborative setting.
- *Case studies:* The course will include case studies of real-world internet and web application projects, enabling students to gain practical insights into the opportunities and challenges associated with internet interconnection.

Practical: Implementation of the concepts discussed in the classroom. This will be lab based where the students will be asked to practically implement the concepts discussed in different units.

Assessment Plan

In-semester (50)

Written exam, assignment, presentation, lab report, mini project report, attendance

End-Semester (50)

objectives, written exam

Reference Materials

Felke-Morris, T. (2015 & 2019). *Web development and design foundations with HTML5* (p. 672). Pearson education limited.

Rosen, L. S. R., & Shklar, L. (2009). Web application architecture: Principles, protocol and practices. Willey.

McDonald, M. (2020). *Web security for developers: real threats, practical defense*. No Starch Press.

Brown, E. (2019). *Web development with node and express: leveraging the JavaScript stack*. O'Reilly Media.

Mathiason, J. (2008). *Internet governance: The new frontier of global institutions*. Routledge.

Additional Resources:

- https://developer.wordpress.org/themes/getting-started/
- https://www.wix.com/blog/how-to-design-a-website?utm_source=google&utm_medium=cpc&utm_campaign=16242175905%5E134377093918&experiment_id=%5E%5E582523585392%5E%5E_DSA&gclid=CjwKCAjw4ZWkBhA4EiwAVJXwqce9ojdSJizAPlSnZceS-gEEIDT3axOeSNBurPbb0CWVpkoocRRFOBoC9TIQAvD_BwE
- https://blog.hootsuite.com/how-to-create-a-youtube-account-channel/
- https://www.youtube.com/watch?v=viAdUNRiYqk
- https://rockcontent.com/blog/email-management/
- https://www.bluehost.com/blog/how-to-build-a-blog-in-5-simple-steps/?irpid=101&clickid=P61C101S570N0B5578A2D4499E0000V135&gclid=CjwKCAjw4ZWkBhA4EiwAVJXwqSfHqZSyX3QqIOsEaiLWV-6eBNmOAQJ6J41ORldehOR4AU9CiwCKFhoCLowQAvD_BwE&gclsrc=aw.ds
- https://www.youtube.com/watch?v=x3c1ih2NJEg

ITEX 214 IT Project Management

Course Title: IT Project Management **Course Code:** ITEX 214
Credit Hours: 2 (1 TH + 1 PR) **Nature:** Theory +Practical

Course Description and Goals

This course provides an introduction to the principles and practices of IT project management. Students will learn key concepts, tools, and techniques necessary to initiate, plan, execute, monitor, control, and close IT projects successfully.

Learning Outcomes

On completion of this course, the students will be able to:

- Understand the fundamental principles and concepts of IT project management.
- Apply project management tools and techniques to IT projects.
- Develop skills in project initiation, planning, execution, monitoring, control, and closure.
- Apply risk management and quality assurance practices in IT project environments.
- Develop effective communication and stakeholder management skills for IT projects.

Content with specific objectives

Specific objectives	Contents
• Gain an understanding of the fundamental principles, methodologies, and best practices used in IT project management. • Explore different project management approaches and frameworks, such as Agile and Waterfall, and learn about industry standards and guidelines for successful project management. • Identify the responsibilities and skills required of an IT project manager, including effective communication, leadership, and stakeholder management. • Discover the key success factors that contribute to the achievement of project goals, such as clear objectives, realistic planning, effective team collaboration, and stakeholder engagement. • Recognize the common challenges faced in IT project management, such as resource allocation, scope creep, risk	**Unit One: Introduction of Project Management [2 hours]** 1.1 Overview of IT project management principles, methodologies, and best practices. 1.2 Role and responsibilities of an IT project manager. 1.3 Key success factors and challenges in IT project management.

mitigation, and managing project dependencies. • Understand the impact of technology changes and dynamic project environments on project management, and explore strategies to overcome challenges for successful project delivery.	
• Understand the importance of project selection and initiation, including criteria and methods used for project evaluation, such as project charters, feasibility studies, and business cases. • Identify and analyze stakeholders in a project, utilizing techniques like stakeholder mapping and analysis to assess their needs, interests, and influence. • Define clear and measurable project objectives, establish scope boundaries and deliverables, and develop a scope statement for effective project planning. • Create a Work Breakdown Structure (WBS) by decomposing project deliverables into work packages, organizing and structuring them using hierarchical numbering and detailed descriptions. • Utilize project scheduling techniques, including Gantt charts, network diagrams (PERT and CPM), critical path analysis, and task duration estimation, to create a project schedule and identify dependencies.	**Chapter 2: Project Initiation and Planning (8 hours)** 2.1 Project Selection and Initiation (1 hour) i. Importance of project selection and initiation ii. Criteria and methods for project selection iii. Project charters, feasibility studies, and business cases 2.2 Stakeholder Identification and Analysis (2 hours) i. Understanding stakeholders and their roles in projects ii. Techniques for identifying stakeholders iii. Stakeholder mapping and analysis iv. Assessing stakeholder needs, interests, and influence v. Managing stakeholder expectations 2.3 Defining Project Objectives and Scope (2 hours) i. Defining clear and measurable project objectives ii. Establishing project scope boundaries and deliverables iii. Developing a scope statement iv. Scope validation and control 2.4 Work Breakdown Structure (WBS) Development (2 hours) i. Introduction to Work Breakdown Structure (WBS) ii. Decomposing project deliverables into work packages iii. Organizing and structuring the WBS iv. Hierarchical numbering and work package descriptions 2.5 Project Scheduling Techniques (1 hour) i. Overview of project scheduling ii. Gantt charts and their applications iii. Network diagrams (PERT and CPM) iv. Critical path analysis v. Task duration estimation and dependency identification

	vi. Creating a project schedule
• Understand the importance of effective team formation and management in IT projects, including team roles, responsibilities, dynamics, and conflict resolution strategies. • Explain the significance of communication in IT projects, including communication planning, stakeholder engagement, and effective communication channels and tools. • Identify and assess project risks, develop risk mitigation and contingency plans, and monitor and control project risks throughout the project lifecycle. • Explain change management processes, including change request evaluation, change control boards, and configuration management, and understand the impact of changes on project scope, schedule, and resources. • Understand the importance of quality assurance and control in IT projects, define quality standards and metrics, and apply quality planning and control activities. • Apply team building, motivation techniques, and stakeholder management strategies in IT projects. • Develop effective project communication plans, including reporting and status updates, and manage project documentation and knowledge sharing. • Evaluate testing and quality assurance techniques and analyze	**Unit Three: Project Execution and Control (10 hours)** 3.1 Team Formation and Management in IT Projects (2 hours) i. Importance of effective team formation and management in IT projects ii. Team roles, responsibilities, and dynamics iii. Team building and motivation techniques iv. Conflict resolution and communication within project teams 3.2 Project Communication and Reporting (2 hours) i. Importance of communication in IT projects ii. Communication planning and stakeholder engagement iii. Effective communication channels and tools iv. Project reporting and status updates v. Managing project documentation and knowledge sharing 3.3 Risk Management in IT Projects (2 hours) i. Understanding project risks and their impacts ii. Risk identification, assessment, and prioritization techniques iii. Developing risk mitigation and contingency plans iv. Monitoring and controlling project risks v. Communication and stakeholder involvement in risk management 3.4 Change Management and Configuration Management (2 hours) i. Change management processes and procedures. ii. Change request evaluation and approval iii. Change control boards and change log management iv. Configuration management and version control v. Impact of changes on project scope, schedule, and resources 3.5 Quality Assurance and Control in IT Projects (2 hours) i. Importance of quality assurance and control in IT projects ii. Defining quality standards and metrics iii. Quality planning and quality control activities iv. Testing and quality assurance techniques

strategies for continuous improvement and capturing lessons learned from project experiences.	v. Continuous improvement and lessons learned
• Understand the importance of proper project closure and the key activities involved in successfully closing a project. • Create and produce the necessary deliverables and artifacts during project closure, ensuring all documentation is finalized and obtaining sign-off from stakeholders. • Organize and archive project documentation for future reference, ensuring easy retrieval and knowledge transfer within the project team. • Analyze and evaluate project success and performance using appropriate methods, comparing project outcomes against defined objectives and metrics. • Conduct lessons learned sessions to capture valuable insights from the project and implement improvements based on the lessons learned to enhance future project outcomes. • Recognize the significance of post-project evaluation and apply strategies for continuous improvement in project management practices.	**Unit Four: Project Closure and Lessons Learned (4 hours)** 4.1 Project Closure Activities and Deliverables (1 hour) i. Importance of proper project closure ii. Key activities and tasks involved in project closure iii. Deliverables and artifacts to be produced during project closure iv. Finalizing project documentation and obtaining sign-off 4.2 Project Documentation and Knowledge Transfer (1 hour) i. Documentation requirements throughout the project lifecycle ii. Organizing and archiving project documentation iii. Knowledge transfer strategies and techniques iv. Capturing lessons learned and best practices 4.3 Post-Project Evaluation and Lessons Learned (2 hours) i. Importance of post-project evaluation ii. Methods for evaluating project success and performance iii. Analyzing project outcomes against objectives and metrics iv. Conducting lessons learned sessions and capturing insights v. Implementing improvements based on lessons learned
Theory	**Total contact hours:24**
Case studies	**Total Contact hours:6**

Practical Works (But not limited to):

While this course does not include lab work, it offers an opportunity to incorporate case studies. Case studies provide real-world examples that allow students to analyze and apply the principles and concepts learned in IT project management. By studying and discussing actual projects and their outcomes, students can gain a deeper understanding of the challenges, decision-making processes, and best practices in IT project management. Case studies serve as

valuable learning tools, encouraging critical thinking, problem-solving, and decision-making skills in the context of real-world scenarios.

3. Healthcare.gov Website Launch: This case study focuses on the launch of the healthcare marketplace website in the United States. It explores the challenges faced in managing the complex IT project, including requirements gathering, system integration, testing, and stakeholder coordination.
4. SAP ERP Implementation at Nike: This case study examines Nike's implementation of an Enterprise Resource Planning (ERP) system by SAP. It delves into the challenges faced in aligning business processes with the new system, data migration, user training, and change management.
5. The FBI's Virtual Case File System: This case study highlights the failed implementation of the FBI's Virtual Case File (VCF) system, which aimed to modernize the agency's case management capabilities. It explores issues related to requirements management, technology selection, vendor management, and project governance.
6. Apple's iPhone Software Update: This case study focuses on a software update release by Apple for its iPhone devices. It analyzes the challenges faced in managing the software development process, ensuring backward compatibility, user experience considerations, and handling unforeseen issues during the update rollout.
7. British Airways' Terminal 5 Project: This case study examines the construction and IT implementation of Terminal 5 at Heathrow Airport by British Airways. It explores the complexities of integrating various IT systems, baggage handling, passenger services, and communication networks to ensure a smooth transition and minimize disruption.
8. Denver Airport Baggage Handling System: The implementation of an automated baggage handling system at Denver International Airport faced significant technical and logistical challenges, leading to delays, cost overruns, and ultimately, the abandonment of the project.
9. National Health Service (NHS) IT System: The NHS in the United Kingdom embarked on a massive IT system overhaul, aiming to create a centralized electronic patient record system. The project faced issues such as scope creep, budget overruns, and lack of user acceptance, resulting in its eventual termination after spending billions of pounds.
10. Hershey's ERP Implementation: Hershey's, the renowned chocolate manufacturer, faced a major setback when its ERP implementation resulted in problems with order fulfillment and inventory management. The system issues during the crucial Halloween season led to significant revenue loss and damaged customer relationships.
11. Queensland Health Payroll System: The implementation of a new payroll system for Queensland Health in Australia encountered extensive delays, cost overruns, and inaccuracies in employee payments. The project's failure resulted in financial losses and disrupted operations for the healthcare organization.
12. UK Ministry of Defense (MoD) Future IT System: The MoD's attempt to modernize its IT infrastructure faced challenges in requirements management, vendor selection, and project governance. The project was ultimately scrapped after significant cost overruns and failure to meet objectives.

13. The Nite Rest Hotel Management System is a case study that highlights the challenges faced in the implementation of an IT project for managing hotel operations. The case study focuses on the failure of the hotel management system and the lessons learned from the experience.

Major Teaching and Learning Strategy

There are two groups of teaching strategies used in this course. The first group comprises of general teaching strategies that can be used with most units. This course can be taught to a) provide a thorough knowledge of project management b) to impart knowledge of formal project management procedures. c) To provide an example of how to use project management concepts at workplace. The specific teaching method for this course will be class discussion with IT Project Management case studies.

Assessment Plan

a) In-semester (50 marks)

a. Internal – I

b. Internal – II

c. Assignment

b) End-semester (50 marks)

a. MCQ – 10

b. Subjective – 40

Text and Reference Books

- Kathy S. K. (2019). *Information technology project management* (9th ed.). Cengage Learning.
- Project Management Institute. (2021). *Project management institute's a guide to the project management body of knowledge (PMBOK® Guide)*. Project Management Institute.