

# Rajarshi Janak University

Faculty of Humanities and Law

Office of the Dean



## Bachelor of Computer Application (BCA)

Second Semester Syllabus

Effective from the Academic Batch of 2025

Sr. No.	Course Code	Course Title	Credit hours
1	BCASO111	The art of social engagement and Technology	3
2	BCACS112	UI/UX using UML	3
3	BCAMT113	Mathematics – II	3
4	BCACS114	OOP in Java	3
5	BCACS115	Microprocessor and Computer Architecture	3
		<b>Total Credit Hours</b>	<b>15</b>

Course Title: **The Art of Social Engagement and Technology**

Course no: **BCA SO111**

Year / Semester: **I/II**

Nature of course: **48 L.H. (Theory)**

Credit hours: **3**

Full Marks: **60+40**

Pass Marks: **24+16**

---

### **Course Description**

The course "*The Art of Social Engagement and Technology*" is an interdisciplinary academic course designed for Bachelor of Computer Applications (BCA) students with a primary foundation in Sociology and Sociological Perspectives. The course examines the relationship between society, culture, communication, human interaction, digital transformation, and technological systems in the contemporary world. In the digital age, technological advancement has significantly transformed social behavior, communication patterns, identity formation, education, governance, business interaction, political participation, and everyday social life. The course introduces students to sociological imagination, digital sociology, cyber culture, social interaction, online communities, digital citizenship, virtual identity, social media engagement, digital inequality, surveillance, algorithmic influence, artificial intelligence ethics, and responsible technological innovation. It also explores how technology shapes social institutions, family relations, cultural practices, social participation, and power relations in both Nepalese and global contexts. Special emphasis is placed on social engagement as a process of meaningful human interaction, civic participation, collaborative communication, community involvement, emotional intelligence, cultural sensitivity, and ethical digital participation. The course further integrates sociological theories with contemporary technological realities to help students become socially aware, ethically responsible, and professionally competent technology practitioners capable of promoting inclusive, human-centered, and socially responsible technological development.

### **Course Objectives**

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

- Explain the relationship between society, technology, and social engagement from sociological perspectives.
- Understand the role of social institutions, culture, norms, and values in shaping technological behavior.
- Analyze the impact of digital technology on communication, identity, social interaction, and social change.
- Evaluate the sociological implications of social media, artificial intelligence, surveillance systems, and digital platforms.
- Understand digital citizenship, cyber culture, virtual communities, and responsible online behavior.
- Examine issues of digital inequality, exclusion, algorithmic bias, and technological power relations.
- Apply sociological theories in understanding technology-driven social transformation.
- Develop interpersonal communication skills, collaborative engagement, emotional intelligence, and community participation.
- Analyze Nepalese digital society and the social impact of technological change in Nepal and South Asia.
- Promote socially responsible, inclusive, ethical, and human-centered technological practices.

## Course Contents

- Unit 1: Foundations of Sociology and Technology** **LH 7**
- Nature, scope, and subject matter of sociology; its relevance to computer applications.
  - Basic sociological concepts: Society, culture, community, socialization, norms, values, status, roles, institutions.
  - Major sociological perspectives: Functionalism, Conflict theory, Post modernist theory.
  - Introduction to Digital Sociology and Applied Sociology.
- Unit 2: Technology, Culture and Social Interaction** **LH 7**
- Meaning, elements and characteristics of culture
  - Cultural diffusion and cultural change through technology
  - Digital socialization and cyber culture
  - Digital communication and interpersonal relations and online identity
  - Technology and changing family, friendship, and relationship patterns
- Unit 3: Inequality, Power and Digital Divides** **LH 8**
- Social stratification (class, caste, gender, ethnicity) in digital contexts.
  - Digital divide: Access, skills, usage, participation, and outcomes in Nepal and globally
  - Power, surveillance, platform capitalism, and algorithmic bias
  - Marginalization, inclusion, and representation in digital society
  - Nepalese digital society:
    - Urban-rural digital gaps
    - Social media and Nepalese youth
    - Cybercrime and misinformation in Nepal
    - E-governance and digital transformation in Nepal
- Unit 4: Ethics, Artificial Intelligence and Future Society** **LH 8**
- Meaning and importance of ethics in technology and digital society
  - Ethical frameworks: Utilitarianism, Deontology, Virtue Ethics, Care Ethics
  - Applied AI ethics: Fairness, accountability, transparency, privacy, and data protection
  - Inclusive technology, design justice, and ethical technological practices
  - Opportunities and challenges of AI in Nepal and South Asia
- Unit 5: Social Media and Social Engagement** **LH 10**
- Meaning, nature and dimensions and importance of social engagement
  - Technology as medium and mediator of engagement
  - Social networking, participatory culture, and digital interaction
  - Technology and changing relationship patterns
  - Theoretical perspective: Herbert Blumer and Symbolic Interactionism, Maxwell McCombs and Agenda Setting Theory, Manuel Castells and Network Society
- Unit 6: Research Methods in Sociology for Digital Contexts** **LH 8**
- Meaning and importance of sociological research in digital society
  - Types of research methods: Qualitative, quantitative, and mixed methods
  - Methods and techniques: Survey research, ethnography, digital ethnography, interviews, questionnaires, content analysis, focus groups, participant observation, and social network analysis

- d) Data collection using digital tools and platforms (Google Forms and ethical social media analysis)
- e) Basic data processing, coding, thematic analysis, statistical interpretation, and visualization techniques (charts, graphs, and network diagrams)

**Suggested Readings:**

- Bell, D. (1973). *The coming of post-industrial society: A venture in social forecasting*. Basic Books.
- Blumer, H. (1969). *Symbolic interactionism: Perspective and method*. University of California Press.
- Dignum, V. (2019). *Responsible artificial intelligence: How to develop and use AI in a responsible way*. Springer.
- Giddens, A., & Sutton, P. W. (2021). *Sociology* (9th ed.). Polity Press.
- Kozinets, R. V. (2020). *Netnography: The essential guide to qualitative social media research* (3rd ed.). Sage Publications.
- McCombs, M. E., & Shaw, D. L. (1972). The agenda-setting function of mass media. *Public Opinion Quarterly*, 36(2), 176-187.
- Neuman, W. L. (2014). *Social research methods: Qualitative and quantitative approaches* (7th ed.). Pearson.
- Slack, J. D., & Wise, J. M. (2015). *Culture + technology: A primer* (2nd ed.). Peter Lang.
- Tegmark, M. (2017). *Life 3.0: Being human in the age of artificial intelligence*. Knopf.
- Zuboff, S. (2019). *The age of surveillance capitalism: The fight for a human future at the new frontier of power*. PublicAffairs.

Course Title: **UI/UX Design with Emerging Technologies**

Course No: **BCACS112**

Year/Semester: **I/II**

Nature of course: **Theory + Lab**

Credit Hours: **3**

Full Marks: **60+20+20**

Pass Marks: **24+08+08**

### **Course Objectives**

After completing this course, students will be able to understand the fundamental concepts and importance of UI/UX design in digital systems. They will apply user-centered design principles and UX research methods to develop user-friendly web and mobile interfaces. Students will gain practical skills in creating wireframes, mockups, and interactive prototypes using modern UI/UX tools. The course also enables students to conduct usability testing and explore emerging technologies such as VUI, conversational UI, wearable interfaces, and AI-assisted design tools.

### **Objectives**

- To understand the fundamental concepts and importance of UI/UX design in digital systems.
- To apply user-centered design principles and UX research methods for creating user-friendly interfaces.
- To develop practical skills in wire framing, mockup creation, and interactive prototyping using tools like Figma and Adobe XD.
- To design responsive, accessible, and visually effective web and mobile user interfaces.
- To evaluate and improve interface usability while exploring emerging technologies such as conversational UI, wearable interfaces, AI-assisted design, and AR-aware UI concepts.

### **Course Contents**

#### **Unit 1: Introduction to UX and UI Design**

**4 L.H.**

1.1 Fundamentals of UX and UI

1.2 Difference between UX and UI

1.3 Roles and responsibilities of UX and UI designers

1.4 UX principles: usability, accessibility, simplicity

1.5 Core disciplines of UX

- User research
- Content strategy
- Information architecture
- Interaction design
- Visual design
- Usability evaluation

1.6 Types of user interfaces

- CLI
- GUI
- VUI
- Menu-driven interface
- NLP-based interface

1.7 Characteristics of good UX/UI design

1.8 UX/UI design tools: Figma, Adobe XD, Sketch

## **Unit 2: User Interaction Design**

4 L.H.

2.1 UX design process and user-centered design

2.2 Mind mapping techniques

2.3 UX research methods

- User interviews
- Surveys
- Competitive analysis

2.4 Creating user personas

2.5 User journey mapping

2.6 Ideations techniques

- Mood boards
- Brainstorming
- Sketching

## **Unit 3: User Interface Design**

6 L.H.

3.1 Graphical and web user interfaces

3.2 Interaction styles

- Command line
- Menu selection
- Form fill-in
- Direct manipulation
- Anthropomorphic interfaces

3.3 Principles of UI design

3.4 Graphical User Interface (GUI) concepts

3.5 UI design process

3.6 Human factors in interface and screen design

3.7 Technological considerations in interface design

## **Unit 4: UI Components**

12 L.H.

4.1 System menus and their functions

4.2 Menu formatting principles

- Consistency
- Display and presentation
- Organization and complexity
- Item arrangement, ordering, and grouping

4.3 Types of menus

- Menu bar
- Pull-down menu
- Cascading menu
- Popup menu
- Tear-off menu
- Iconic menu

4.4 Window selection and components

4.5 Window presentation styles

- Tiled windows
- Overlapping windows
- Cascading windows

#### 4.6 Types of windows

- Primary windows
- Secondary windows
- Dialog boxes

#### 4.7 Screen-based controls

- Buttons and toolbars
- Text entry controls
- Read-only controls
- Radio buttons
- Checkboxes
- Palettes
- List boxes
- List view controls
- Drop-down and popup list boxes

#### 4.8 Other operable controls

- Sliders
- Tabs
- Date pickers
- Tree views
- Scroll bars

#### 4.9 Selection of appropriate controls

#### 4.10 Creating meaningful graphics, icons, and images

### Unit 5: UI Design Considerations

6 L.H.

#### 5.1 Page layout and responsive design

#### 5.2 Color schemes and typography

#### 5.3 Font selection and screen size considerations

#### 5.4 Interactive elements in interface design

#### 5.5 Visual hierarchy principles

- Alignment
- Color
- Contrast
- Proximity
- Size
- Texture
- Time

#### 5.6 Navigation types

- Global navigation
- Utility navigation
- Associative navigation
- Inline navigation

#### 5.7 Navigation models

- Hub and spoke
- Fully connected
- Multilevel/tree navigation
- Stepwise navigation

- Pyramid navigation
- Flat navigation

6 L.H.

## **Unit 6: Wire-framing and Prototyping**

### **Topics:**

- 6.1 Wireframes and mockups
- 6.2 Types of prototyping
  - Low-fidelity prototyping
  - High-fidelity prototyping
  - Interactive prototyping
- 6.3 UX storyboarding and mockups
- 6.4 Software prototyping
- 6.5 Transitions and animations in prototypes
- 6.6 Creating clickable prototypes

6 L.H.

## **Unit 7: Design Evaluation**

### **Topics:**

- 7.1 Formative and summative evaluation
- 7.2 Usability testing
  - Moderated testing
  - Unmoderated testing
- 7.3 Analysis of usability test results
- 7.4 Expert analysis and user participation
- 7.5 Iterative evaluation paradigms
- 7.6 DECIDE evaluation framework
- 7.7 Heuristic evaluation
- 7.8 Task analysis and performance metrics

## **Unit 8: Advanced Techniques – VUI and NLP-Based UI**

4 L.H.

### **Topics:**

- 8.1 Command and control interfaces vs conversational UI
- 8.2 Personas, avatars, actors, and video game interfaces
- 8.3 Speech recognition technology
- 8.4 Dialog management systems
- 8.5 Designing interfaces for wearable devices
- 8.6 Introduction to AR-aware UI concepts

## **Practical/Laboratory Works**

### **Practical 1: Introduction to UI/UX Tools**

- Installation and setup of Figma/Adobe XD
- Creating simple screens such as login and home pages
- Converting sketches into wireframes
- Exploring auto-layout and alignment features

### **Practical 2: UX Research and User Flows**

- Conducting short user interviews
- Performing surveys and competitive analysis

- Creating user personas
- Developing user journey maps
- Sketching low-fidelity UI flows

### **Practical 3: Interface Design and Components**

- Designing core screens such as login, home, and profile pages
- Applying color, typography, and layout principles
- Organizing UI controls into reusable components
- Creating simple design tokens
- Adding basic micro-interactions and animations

### **Practical 4: Prototyping and Evaluation**

- Converting UI screens into clickable prototypes
- Conducting usability testing with peers
- Creating heuristic evaluation checklists
- Evaluating peer designs
- Using AI-assisted design features for faster prototyping

### **Practical 5: Emerging UI Technologies and Final Project (3 Weeks)**

- Designing a simple conversational/VUI application
- Creating wearable-style or AR-aware interface screens

### **Final Project Requirements:**

- Problem identification and statement
- Creation of 1–2 user personas
- User journey mapping
- Designing 3–5 core UI screens using Figma/Adobe XD
- Developing an interactive prototype
- Preparing a one-page project reflection report
- Delivering a short presentation

### **References**

1. Designing User Experience – Benyon, David. *Designing User Experience: A Guide to HCI, UX and Interaction Design*. 4th Edition, Pearson Education, 2019.
2. Designing Interfaces – Tidwell, Jenifer; Brewer, Charles; and Valencia, Aynne. *Designing Interfaces*. 3rd Edition, O'Reilly Media, 2020.
3. Human Computer Interaction – Malik, D. S. and Gupta, Pooja. *Human Computer Interaction*. 1st Edition, University Science Press, India, 2018.
4. Human Computer Interaction – Jyoti, Anirban and Roy, Soma. *Human Computer Interaction*. 1st Edition, BPB Publications, India, 2021.
5. Web Designing and Development – Jain, Satish and Shashi Singh. *Web Designing and Development*. Latest Edition, BPB Publications, India.

Course Title: **Mathematics - II**

Course no: **BCAMT113**

Year / Semester: **I/II**

Nature of course: **48 L.H. (Theory + Lab)**

Credit hours: **3**

Full Marks: **60+20+20**

Pass Marks: **24+08+08**

---

### **Course Description**

The Mathematics II course is designed to provide students with a strong foundation in calculus, computational methods, linear algebra, and abstract algebra with applications in computer science and computational problem solving. The course covers fundamental concepts of limits, continuity, differentiation, and integration along with their practical applications in optimization, curve analysis, business mathematics, and geometrical computations. It also introduces numerical and computational techniques such as linear programming, solution of linear equations, and numerical methods using tools like MATLAB and GNU Octave. In addition, the course explores algebraic structures including groups, rings, fields, and vector spaces to develop analytical, logical, and mathematical reasoning skills essential for computer applications and advanced studies in computing and engineering.

### **Objectives**

- To understand the fundamental concepts of calculus including limits, continuity, differentiation and integration
- To apply mathematical techniques for solving real world and business related problems and profit maximization
- To develop computational problem solving skills using numerical methods and matrix operations
- To gain practical knowledge of mathematical computing using MATLAB and GNU Octave
- To understand algebraic structures and vector spaces used in computer application and advanced mathematics.

### **Course contents:**

#### **Unit-1 Limits and continuity**

**[8 L.H.]**

##### **1.1 Limit of a function**

1.1.1 Indeterminate Forms

1.1.2 Algebraic properties of limit (without proof)

1.1.3 Theorem on limits of algebraic and Algebraic and Transcendental function

##### **1.2 Continuity of a function**

1.2.1 Types of discontinuity

1.3 Exercises on evaluation of limits and test of continuity.

### **Lab**

Lab 1: Evaluation of Limits Numerically and Graphically

Lab 2: Testing Continuity of Functions

## Unit-2 Differentiation

[5 L.H.]

- 2.1 Introduction to Differentiation
- 2.2 Techniques of differentiation
- 2.3 Derivative of Algebraic and Transcendental function
- 2.4 Higher order derivative

### Lab

Lab 3: Numerical and Symbolic Differentiation

Lab 4: Higher Order Derivatives

## Unit-3 Application of Differentiation

[8 L.H.]

- 3.1 Derivative as slope and rate measure
  - 3.1.1 Use of L'Hospital rule to evaluate limit
  - 3.1.2 Monotonicity and concavity of curves of function,
  - 3.1.3 Stationary and inflectional points
  - 3.1.4 Maximum and minimum values
  - 3.1.5 The mean value theorem
- 3.2 Cost, revenue and profit functions
  - 3.2.1 Average cost & marginal cost
  - 3.2.2 Average revenue & marginal revenue
  - 3.2.3 Profit maximization under perfect competition and profit maximization under monopoly.

### Lab

Lab 5: Maxima and Minima of Function

Lab 6: Plotting Monotonicity and concavity

Lab 7: Business application – Profit Maximization

## Unit-4 Anti-derivative and its applications

[8 L.H.]

- 4.1 Introduction
  - 4.1.1 Indefinite integral
  - 4.1.2 Techniques of integrations
  - 4.1.3 Fundamental theorem of integral calculus (conceptual)
  - 4.1.4 Definite integral
  - 4.1.5 Double integration (Conceptual)
  - 4.1.6 Evaluation of definite and indefinite integration.
- 4.2 Application of definite integral
- 4.3 Rectification and Quadrature
- 4.4 Area under a curve
- 4.5 Area between the curves

4.6 Volumes and Area of surface of revolution.

**Lab**

Lab 8: Indefinite Integration to compute indefinite integrals

Lab 9: Definite Integration to evaluate definite integrals numerically and symbolically

Lab 10: Area under curve to find the area under curves

Lab 11: Compute volume generated by Revolution of curves

**Unit-5 Computational Method**

**[8 L.H.]**

5.1 Linear programming problem (LPP)

5.1.1 Solution of LPP by graphical

5.1.2 Solution of LPP Simplex methods (up to 3 variables)

5.2 System of linear equation

5.2.1 Consistent and inconsistent system

5.2.2 Solution of system of linear equation by Gauss elimination method, Gauss seidel method, matrix inverse method, row equivalent matrix method.

5.3 Bisection method for solving Non-linear equation.

**Lab**

Lab 12: Solution of Linear Equations using Gauss Elimination to solve systems of linear equation

Lab 13: Gauss Seidel Method to solve linear equation iteratively.

Lab 14: Matrix Inverse Method to solve equations using inverse matrices.

Lab 15: Bisection Method to solve nonlinear equations numerically

Lab 16: Linear Programming Problem using MATLAB to solve LPP using graphical and built in methods

**Unit-6 Group, Rings and Fields**

**[6 L.H]**

6.1 Group

6.1.1 Binary operations

6.1.2 Algebraic structures

6.1.3 Group and subgroup

6.2 Rings and fields.

6.2.1 Definition of Rings

6.2.2 Definition of fields

**Lab**

Lab 17: Verification of group properties to verify closure, associativity, identity and inverse.

Lab 18: Binary operations table for modular arithmetic to generate operation tables for modular arithmetic.

## Unit-7 Vector spaces

[5 L.H.]

- 7.1 Introduction
- 7.2 Vector space with examples
- 7.3 Subspaces with examples
- 7.4 Linear combination of vectors
- 7.5 Linear dependence and independence of vectors
- 7.6 Basis and dimension of vector space.

### Lab

Lab 19: Linear combination of vectors to verify linear combinations.

[Note: Mathematica, Matlab/Octave, Python should be used as tools to perform the lab work mentioned above.]

### Text Book/ Reference Books

1. James Stewart, Calculus Early Transcendentals, cengage Learning, Delhi, India 2012 (9<sup>th</sup> edi)
2. Thomas, G.B., Finney, R.S., "Calculus with Analytic Geometry", Addison Wesley 9<sup>th</sup> Edition.
3. Budnick, F.S. Applied Mathematics for Business Economics and Social Science, McGraw-Hill Ryerson Ltd.
4. I.S.J. Leon, Linear Algebra with Application, 9<sup>th</sup> ed. Pearson, 2015

**Course Title: Object Oriented Programming in JAVA**

**Course no: BCACS114**

**Year / Semester: I/II**

**Nature of course: 48 L.H. (Theory + Lab)**

**Credit hours: 3**

**Full Marks: 60+20+20**

**Pass Marks: 24+08+08**

---

### **Course Description**

This course introduces the principles of Object-Oriented Programming (OOP) using Java and focuses on developing problem-solving and application development skills. It covers fundamental Java programming concepts including data types, variables, operators, control structures, methods, arrays, strings, and classes. The course further explores advanced OOP concepts such as inheritance, polymorphism, abstraction, encapsulation, interfaces, packages, constructors, recursion, and inner classes. Students also learn event-driven and GUI programming using AWT and Swing, exception handling, file handling, multithreading, and database programming using JDBC. Practical laboratory exercises are integrated throughout the course to provide hands-on experience in designing and developing Java-based applications.

### **Course Objectives:**

- Understand the fundamental concepts and features of Object-Oriented Programming (OOP).
- Develop Java programs using variables, data types, operators, control structures, methods, arrays, and strings.
- Design and implement classes and objects to model real-world problems.
- Apply OOP principles such as inheritance, polymorphism, abstraction, and encapsulation in Java applications.
- Use constructors, recursion, interfaces, packages, and access control mechanisms effectively.
- Develop event-driven and GUI-based applications using AWT and Swing components.

### **Contents**

#### **Unit 1: Introduction to Programming Concept**

**8 L.H.**

##### **1.1 Introduction of OOP**

- 1.1.1 Definition of OOP approach
- 1.1.2 Features of OOP
- 1.1.3 Limitations of OOP
- 1.1.4 Comparison between Procedural and OOP based language

##### **1.2 Introduction to JAVA**

- 1.2.1 Basics of JAVA programming
  - 1.2.1.1 Datatypes
  - 1.2.1.2 Variables
  - 1.2.1.3 Operators
  - 1.2.1.4 Control structure including selection structure
  - 1.2.1.5 Looping
  - 1.2.1.6 Java methods
  - 1.2.1.7 Overloading
  - 1.2.1.8 Arrays
  - 1.2.1.9 String
  - 1.2.1.10 Comments

### **Lab**

- Basic java program using datatypes, variables, operators and comments
- Control structure and looping

- Java methods and Methods overloading
- Array operations
- String handling

## Unit 2: Class and Objects

8 L.H.

### 2.1 Fundamentals of classes

- 2.1.1 Simple class
- 2.1.2 Creating class instances
- 2.1.3 Adding methods to a class

### 2.2 Calling Functions and Methods

- 2.2.1 Abstraction
- 2.2.2 Encapsulations
- 2.2.3 Using 'this' keyword
- 2.2.4 Constructor
- 2.2.5 Default constructor
- 2.2.6 Parameterized constructor

### 2.3 Calling function by Value

### 2.4 Calling function by reference

### 2.5 Access control

### 2.6 Methods that returns value

### 2.7 Recursion

- 2.7.1 Nested
- 2.7.2 Inner class

### Lab

- Simple class, Object creation, methods and 'this' keyword
- Constructors(Default and Parameterized)
- Abstraction, Encapsulation, and Access control
- Call by value and recursion
- Call by reference, Nested class and Inner class

## Unit 3: Inheritance and Polymorphism

12 L.H.

### 3.1 Inheritance

- 3.1.1 Inheritance in Java
- 3.1.2 Super and Sub class declaration
- 3.1.3 Class overriding
- 3.1.4 Object class

### 3.2 Polymorphism

- 3.2.1 Definition of Polymorphism
- 3.2.2 Method overloading
- 3.2.3 Method overriding
- 3.2.4 Implementation technique

### 3.3 Casting Objects

### 3.4 Abstract class

### 3.5 Interface in Java

### 3.6 Package in java

#### 3.6.1 Util package

#### Lab

- Inheritance, Super Class, Sub Class and Method Overriding
- Polymorphism using Method Overloading and Method Overriding
- Object Casting and Object Class
- Abstract Class and Interface
- Package and Java.util Package

### Unit 4: Event Handling, Applet and Swing

13 L.H.

#### 4.1 Event and GUI Programming

- 4.1.1 Event Handling in Java
- 4.1.2 Event Types
- 4.1.3 Mouse and Key Events
- 4.1.4 GUI Basics
- 4.1.5 Panels
- 4.1.6 Frames

#### 4.2 Layout Managers

- 4.2.1 Flow layout
- 4.2.2 Border layout
- 4.2.3 Grid layout

#### 4.3 GUI Components

- 4.3.1 Buttons
- 4.3.2 Check Boxes
- 4.3.3 Radio buttons
- 4.3.4 Labels
- 4.3.5 Text fields
- 4.3.6 Text area
- 4.3.7 Combo boxes
- 4.3.8 Lists
- 4.3.9 Scroll bars
- 4.3.10 Sliders
- 4.3.11 Windows
- 4.3.12 Menus
- 4.3.13 Dialog boxes

#### 4.4 Applet

- 4.4.1 Definition
- 4.4.2 Applet life cycle
- 4.4.3 Control using applet

#### 4.5 Introduction to Swing

#### 4.6 Exception handling using try and catch method

#### 4.7 I/O programming

- 4.7.1 Text and Binary I/O
- 4.7.2 Binary I/O classes

4.7.3 Object I/O

4.7.4 Random Access File

**Lab**

- GUI Components, Event Handling and Layout Manager
- Mouse event, Key event and different layouts
- Swing components with menu and dialog Box
- Exception handling and file I/O
- Binary I/O, Object I/O, Random Access File and Applet

**Unit 5: Multithreading and Database programming**

**7 L.H.**

**5.1 Multithreading in Java**

5.1.1 Thread Life Cycle and Methods

5.1.2 Runnable Interface

5.1.3 Thread synchronization

**5.2 Database programming using Java**

5.2.1 Database connectivity

5.2.2 Statement set for manipulating Data in database

5.2.3 Result set interface

**Lab**

- Multithreading using Thread class
- Multithreading using Runnable Interface
- Thread synchronization
- Database connectivity and Data Manipulating using JDBC
- ResultSet Interface and Fetching Data from Database

**References**

**Text Books**

- [1] H. Schildt, *Java: The Complete Reference*, 11th ed. New York, USA: McGraw-Hill Education, 2018.
- [2] E. Balagurusamy, *Programming with Java*, 6th ed. Chennai, India: McGraw-Hill Education, 2019.
- [3] P. Radha Krishna, *Object-Oriented Programming through Java*, 2nd ed. New Delhi, India: Universities Press, 2012.

**Reference Books**

- [4] C. S. Horstmann, *Core Java Volume 1 – Fundamentals*, 11th ed. Boston, USA: Pearson Education, 2018.
- [5] K. Sierra and B. Bates, *Head First Java*, 3rd ed. Sebastopol, USA: O'Reilly Media, 2022.
- [6] P. Deitel and H. Deitel, *Java How to Program (Early Objects)*, 11th ed. New York, USA: Pearson, 2017.
- [7] B. Eckel, *Thinking in Java*, 4th ed. Upper Saddle River, USA: Prentice Hall, 2006.

**Course Title:** Microprocessor and Computer Architecture

**Course Code:** BCACS115

**Year / Semester:** I/II

**Nature of Course:** Theory + Lab

**Credit Hours:** 3

**Full Marks:** 60+20+20

**Pass Marks:** 24+8+8

### Course Description

This course introduces undergraduate students to the fundamentals of microprocessors and computer architecture, focusing on Intel 8085 and 8086 microprocessors, assembly language programming, CPU organization, memory hierarchy, pipelining, I/O systems, and arithmetic operations. The course integrates theoretical foundations with laboratory experimentation using simulators and microprocessor trainer kits to develop both programming and architectural understanding.

### Course Objectives

Upon completion of this course, students will be able to:

- Explain the architecture and organization of microprocessor-based systems.
- Analyze the functional units of Intel 8085 and Intel 8086 microprocessors.
- Develop and debug assembly language programs using addressing modes and instruction sets.
- Interpret instruction cycles, machine cycles, timing diagrams, and control signals.
- Explain CPU organization, datapath operations, and micro-operations.
- Compare hardwired and microprogrammed control units.
- Differentiate RISC and CISC architectures.
- Explain memory hierarchy, DMA, and I/O organization.
- Analyze instruction pipelining, hazards, and processor performance.
- Implement arithmetic algorithms and microprocessor-based applications using simulators and trainer kits.

### Course Contents

#### Unit 1

#### Introduction to Microprocessors and Computer Systems

[3 Hrs]

- Definition and evolution of microprocessors
- Microprocessor vs microcontroller
- Components of a microprocessor:
  - ALU
  - Registers
  - Control Unit
  - Timing Unit
- System buses:
  - Address Bus
  - Data Bus
  - Control Bus
- Bus organization and system architecture
- Applications of microprocessors in embedded and computing systems

#### Practical

- Familiarization with 8085 trainer kit and simulator environment
- Identification of processor components and buses

## Unit 2

### Intel 8085 Microprocessor and Assembly Language Programming

[12 Hrs]

- Architecture and Organization
  - Functional block diagram of 8085
  - Pin configuration
  - Registers and register organization
  - Flags and flag register
  - Multiplexed address/data bus
  - Timing and control signals
  - Interrupts (Introductory concepts)
  
- Instruction Execution
  - Addressing modes
  - Instruction cycle and machine cycle
  - Opcode fetch, memory read/write cycles
  - Timing diagrams:
    - MOV
    - MVI
    - LDA
    - STA
  
- Instruction Set
  
- Data Transfer Instructions  
MOV, MVI, LDA, STA, LXI, LDAX, STAX, XCHG
  
- Arithmetic Instructions  
ADD, ADI, ADC, SUB, SUI, SBB, INR, DCR, INX, DCX
  
- Logical Instructions  
ANA, ANI, ORA, ORI, XRA, XRI, CMA, CMP
  
- Branching Instructions  
JMP, JNZ, JZ, JC, JNC
  
- Assembly Programming
  - Looping and branching
  - Bitwise operations
  - Array processing
  - Searching and sorting concepts
  - 8-bit and 16-bit arithmetic programs
  - Multiplication and division algorithms

#### Practical

- Assembly programming using trainer kit and simulator
- Demonstration of addressing modes

- Programs for:
  - Arithmetic operations
  - Logical operations
  - Array search
  - Looping and branching
- Timing diagram analysis using simulator tools

### Unit 3

#### 8086 Microprocessor

[4 Hrs]

- Architecture of 8086
- Bus Interface Unit (BIU)
- Execution Unit (EU)
- Register organization and flags
- Memory segmentation
- Segment registers
- Introduction to instruction pipelining
- Comparison between 8085 and 8086 architecture

#### Practical

- Simulation of segmented memory access
- Demonstration of instruction prefetch and pipelining using 8086 simulator

### Unit 4:

#### Basic Computer Architecture and Datapath Design

[5 Hrs]

- Stored program organization
- Register transfer concepts
- Computer registers
- Common bus system
- Instruction formats and instruction codes
- Micro-operations:
  - Arithmetic
  - Logic
  - Shift operations
- Datapath concept and data movement between registers and ALU

#### Practical

- Implementation of arithmetic, logic, and shift micro-operations using simulator
- Register transfer and datapath tracing exercises

### Unit 5:

#### Control Unit Organization

[5 Hrs]

- Hardwired control unit
- Microprogrammed control unit
- Control memory
- Address sequencing
- Microinstruction formats
- Comparison of hardwired vs microprogrammed control

### Practical

- Simulation of control signal sequencing
- Microinstruction format analysis using control unit simulator

### Unit 6

#### Central Processing Unit Organization

[5 Hrs]

- General register organization
- Stack organization
- Stack operations and subroutine concepts
- Instruction formats:
  - Three-address
  - Two-address
  - One-address
  - Zero-address
- RISC vs CISC architecture comparison

### Practical

- Stack manipulation and subroutine simulation
- Comparative analysis of RISC and CISC instruction execution

### Unit 7

#### Parallel Processing and Pipelining

[4 Hrs]

- Flynn's classification
- Speedup ratio
- Arithmetic pipeline
- Instruction pipeline
- Pipeline hazards:
  - Data hazards
  - Control hazards
- Branch handling techniques

### Practical

- Pipeline simulation exercises

### Unit 8

#### Computer Arithmetic

[4 Hrs]

- Signed magnitude representation
- 1's complement and 2's complement arithmetic
- Addition and subtraction algorithms
- Booth multiplication algorithm
- Overflow detection concepts

### Practical

- Implementation of arithmetic algorithms using high-level language
- Simulation of Booth multiplication algorithm

## Unit 9

### I/O Organization and Memory Hierarchy

[6 Hrs]

#### I/O Organization

- Peripheral devices
- I/O bus and interface modules
- Isolated I/O
- Memory-mapped I/O
- Interrupt-driven I/O
- Direct Memory Access (DMA)
- DMA controllers and transfer procedure

#### Memory Organization

- RAM and ROM organization
- Address decoding
- Cache memory concepts
- Locality of reference
- Primary vs secondary memory,

#### Practical

- Memory interfacing and address decoding simulation
- Comparison of isolated and memory-mapped I/O

#### Laboratory Requirements

##### Hardware

- 8085 Microprocessor Trainer Kit
- FPGA/Microcontroller development board (optional)

##### Software

- GNUsim8085 / Sim8085
- EMU8086
- VHDL
- Proteus / Multisim
- High-level language compiler (C/Python)

#### Textbooks

- Gaonkar, R. S. (1998). *Microprocessor architecture, programming, and applications with the 8085*. Prentice-Hall, Inc..
- Hall, D. V. (1986). *Microprocessors and interfacing: programming and hardware*. McGraw-Hill, Inc..
- Mano, M. M. (1993). *Computer system architecture*. Prentice-Hall, Inc..