Course Description

6th Semester:

Course Name: Computer Organization and Architecture **Course Code: 18B11CI414**
L-T-P scheme: 3-1-0 **Credits: 4**

Pre-requisite: Digital System and Microprocessors

18B11CI414: Computer Organization & Architecture	
Course Outcome	**Description**
CO1	Develop the understanding of data representation and digital logic circuits used in the computer system.
CO2	Concepts of Register Transfer Language (RTL) to design data transfer bus, combinational and sequential logic circuits.
CO3	Understand the programming of basic computer system using machine language, assembly language and microinstructions.
CO4	Describe the various architectures of CPU and their designing concepts.
CO5	Memory hierarchy, cache memory, virtual memory and their working principle/performance.

Course Contents:

Unit-1: Digital Logic Circuits - Logic Gates, Boolean Algebra, Map Specification, Combinational Circuits, Flip-Flops, Sequential Circuits, Memory Components, Integrated Circuits. Data Representation - Data Types, Complements, Fixed Point Representations Floating Point Representations, Other Binary Codes, Error Detection Codes.

Unit-2: Register Transfer and Micro operations – Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro operations, Logic Micro operations, Shift Micro operations, Arithmetic Logic Shift Unit. Basic Computer Organization And Design- Instruction Codes, Computer Registers, Computer Instructions, Timing And Control, Instruction Cycle, Memory Reference Instructions, Input-Output And Interrupt, Complete Computer Description, Design Of Basic Computer, Design Of Accumulator Logic.

Unit-3: Programming The Basic Computer - Introduction to Machine Language, Assembly Language, Assembler, Program Loops, Programming Arithmetic And Logic Operations Subroutines, Input-Output Programming. Micro programmed Control-Control Memory, Sequencing Microinstructions, Micro program Example, Design Of Control Unit, Microinstruction Format.

Unit-4: Central Processing Unit – Introduction To CPU, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer And Manipulation, Program Control, Reduced Instruction Set Computer. Pipelining and Vector Processing - Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline.

Unit-5: Computer Arithmetic – Introduction to Computer Arithmetic, Addition and Subtraction, Multiplication algorithms, Division algorithms, floating point arithmetic Operations, Decimal Arithmetic Unit, Decimal Arithmetic Operations. Input-Output Organization - Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes Of Transfer, Priority Interrupt, Direct Memory Access, Input-Output Processor, Serial Communication.

Unit-6: Memory Organization - Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory, Memory Management Hardware. Multiprocessors-Characteristics of Multiprocessors.

Text Books

1. "Computer System Architecture", M. Morris Mano, Third Edition.
2. "Computer Organization and Architecture", William Stalling, Tenth Edition.

Other References:

1. Yu-cheng Liu,Glenn A. Gibson , "The 8086/8088 Family Architecture, Programming & design", Second Edition, PHI.
2. Douglas Hall, "Microprocessors & Interfacing, Programming & Hardware",2nd Edn. Tata McGraw Hill.
3. Kenneth Ayala **"**The 8086 microprocessor programming and Interfacing the PC".
4. Tom Shanley, Protected Mode Software Architecture, Addison-Wesley (1996), ISBN 0-201-55447-X .

Resources

Lecture presentations, assignments and practicals, will be posted on the student resource from time to time. In addition following additional online/downloadable resources will be useful.

- NPTEL Course: Computer architecture and organization, IIT Kharagpur by Prof. Indranil Sengupta, Prof. Kamalika Datta, https://nptel.ac.in/courses/106105163
- Official IA-32 Programmer Reference Manuals online at http://developer.intel.com/design/Pentium4/documentation.htm
- Professor Ralf Brown's Interrupt List online at http://www.ctyme.com/rbrown.htm
- Homepage for H. Peter Anvin's SYSLINUX Project online at http://syslinux.zytor.com/
- Online article: The GNU GRUB Boot Loader by Jaswinder Singh Kohli (Linux Gazette #64, 2001)
- Official Data Sheet for the Intel 8259A Programmable Interrupt Controller (.pdf file-format)

Evaluation Scheme:

Test-1	15 marks
Test-2	25 marks
Test-3	35 marks
Assignments	10 marks
Tutorial	5 marks
Quizzes	5 marks
Total	**100 marks**

Title of Course: Software Engineering **Course Code** 18B11CI612
L-T-P Scheme: 3-0-0 **Credits: 3**

Pre-requisite: Good Knowledge of Computer Programming

Post Course:
Object Oriented Software Engineering, Software Quality Management Objective: To engineer good quality software from its specification

Learning Outcomes

Software Engineering	
Course Outcome	**Description**
CO1	Outline various software models with respect to their needs of the customer requirement and concepts of some modeling language.
CO2	Describe the real world problems using software engineering concepts and tools.
CO3	Develop the software design to meet customer expectations using modeling language.
CO4	Identify and use various cost estimation techniques used in software engineering project management.
CO5	Apply verification and validation techniques on a given software project.
CO6	Demonstrate deployment and basic maintenance skills.

Course Outline:
Interactive Systems, Usability, Introduction to software engineering, Software process models, PSP, TSP Requirement Engineering: Requirement Elicitation, Analysis, Specification, SRS, Formal system development techniques, Analysis and Modeling: Data modeling, Functional modeling, Software Architecture and Design: Data design, Architectural Design Process, SADT, OOAD, function-oriented design

UML: Use case diagram, State diagram, Activity Diagram, Class Diagram, Sequence diagram, Collaboration diagram, Deployment Diagram, Event trace diagram, Design Patterns: Structural Patterns, Behavioral Patterns, Creational Patterns

Software Estimation- Estimating Size, Effort and Cost: Metric for Analysis, Metric for Design, COCOMO model, Putnam Model etc., Implementation and Integration: Coding standard and practices, Top-Down and Bottom–up Approach, Verification and Validation,

Software Testing: Structural testing, functional Testing, Testing Strategies, Test Case design.

Software Maintenance: Types, Cost of Software, maintenance, Software Maintenance Models
CASE Tool Taxonomy: Business Process Engineering tool, Process modeling and management tool, project planning tool, requirement tracking tool, Metric and management tool, documentation tool, system software tool etc.Introduction to software engineering for web and mobile applications.

Teaching Methodology:
This course should be conducted in a highly interactive environment. Students will work on different software projects in small groups. Exercises shall almost exclusively consist of design work and the laboratory shall be a place to develop these designs using CASE tools. As part of lab work there shall be a project to build a specification and convert it into working software using Rational Unified Process. Also, there shall be a testing project. There is a self learning component that shall be announced.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1 & Unit-2
Test-2	25 Marks	Based on Unit-3 to Unit-4 and around 20% from coverage of Test-1
Test-3	35 Marks	Based on Unit-5 to Unit-6 and around 40% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Text Book

1. The Unified Modeling Language Users Guide: Grady Booch, James Rambaugh, Ivar Jacobson, Addision Wesley.
2. Douglas Bell, "Software Engineering for students: a programming approach", 4th Ed Pearson Education, 2005.
3. Dines, Bjorna "Software Engineering: abstraction and Modelling"Vol.1, 2006, Springer Verlag Berlin Heidelberg (206).
4. Cooling Jin, "Software Engineering for real time systems, Addison Wesley.
5. Khoshgoftaar, Taghi M. "Software Engineering with Computational Intelligence".
6. Sommerville, Ian, "Software Engineering", 8th Edition, Pearson Education Ltd.
7. Pressman S. Roger, "Software Engineering: A practitioner's Approach", 7th Edition, McGraw Hill.

Web References:

1. https://onlinecourses.nptel.ac.in/noc20_cs68/preview
2. https://online.visual-paradigm.com/
3. https://www.coursera.org/learn/introduction-to-software-engineering

Title of Course: Minor Project-2 **Course Code: 18B91CI691**
Course Credits: 3

Course Learning Outcome:

After successful completion of this course student will be able to:

Course Outcome	Description
CO1	Analyze chosen literature addressing real world research problem to identify the requirements
CO2	Build technical report detailing the software specification, design, test plan, and implementation details.
CO3	Build a practicable solution for the research problem
CO4	Evaluate results to test the effectiveness of the proposed solution
CO5	Develop effective communication skills for presentation of project related activities

Syllabus:

A project to be developed based on one or more of the following concepts.

Project based learning: Each student in a group of 3-4 will have to develop a Minor Project based on different engineering concepts. The students can opt any real-world application for the implementation of Minor Project. The students have to implement the real world problem using any open-source programming language. Project development will enhance the knowledge and employability of the students in IT sector.

Evaluation scheme:

Exam	Marks
P1	10 marks
P2	15 marks
P3	30 marks
Term paper	20 marks
Guide marks	25 marks (continuous evaluation-15, documentation-10)
Total	**100 marks**

Title of Course: **Computer Organization and Architecture Lab** **Course Code: 18B17CI474**
L-T-P scheme: 0-0-2 **Credit: 1**

Prerequisite: Students must have knowledge of Digital Systems and Microprocessors (DSM) subject.

Objective:

1. To acquire the generic hardware development skill through various stages of designing.
2. To design and verify computer system digital circuits in VHDL.
3. To ensure the quality of hardware through various levels of verifications with Xilinx software and ISim simulator.

Learning Outcomes:

Course Outcome	**Description**
CO1	Designing of basic building blocks of a computer system.
CO2	Implementation of basic adder-subtractor units.
CO3	Learn to design the ALU of a computer system.
CO4	Understand the designing of computer data bus architecture.
CO5	Memory (RAM/ROM) system designing.
CO6	Designing of sequential logic circuits for a computer system.

Course Content:
Unit-1; Introduction to VHDL and Xilinx ISE Software.
Unit-2: Design of All-in-One logic gate circuits.
Unit-3: Design of 4-bit adder-subtractor circuits.
Unit-4: Design of combinational logic circuits.
Unit-5: Design of multiplexer based N-bit common bus system, logic system, and shift system.
Unit-6: Design of Arithmetic Logic Shift Unit (ALU).
Unit-7: Design of registers and counters.
Unit-8: Design of ROM and RAM.

Evaluation Scheme:

Exams		**Marks**	**Coverage**
P-1		15 Marks	**Based on Lab Exercises: 1-4**
P-2		15 Marks	**Based on Lab Exercises: 5-8**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Learning Resources:

Soft copies of study material and lab exercises of Computer Organization and Architecture Lab are made available on the JUET server.

Text Book

1. "VHDL Primer"; J. Bhasker , 3 rd Edition.
2. "Computer System Architecture" by M Morris Mano, Third Edition.

Other References:

3. The student's guide to VHDL"; Peter. J. Ashenden, 2nd Edition.
4. Computer System Organization and Architecture: Designing for Performance" by W Stallings, Seventh Edition, Prentice Hall, 2006. ISBN: 0-13-185644-8.

Resources

Lecture presentations, assignments and practicals, will be posted on the student resource from time to time. In addition following additional online/downloadable resources will be useful.

- Xilinx Documentation Portal: https://docs.xilinx.com/r/en-US/xilinx-documentation-portal
- NPTEL Course: Computer architecture and organization, IIT Kharagpur by Prof. Indranil Sengupta, Prof. Kamalika Datta, https://nptel.ac.in/courses/106105163
- Virtual Lab: http://vlabs.iitkgp.ac.in/coa

Title of Course: Software Engineering Lab **Course Code: 18B17CI672**
L-T-P scheme: 0-0-2 **Course Credit: 1**
Prerequisite: Students must have already registered for the course, "Software Engineering".

Objectives: Students will be capable to acquire the generic software development skill through various stages of software life cycle. He will also be able to ensure the quality of software through software development with various protocol based environment.

Software Engineering Lab	
Course Outcome	**Description**
CO1	Outline various software models with respect to their needs of the customer requirement and concepts of some modelling language.
CO2	Describe the real world problems using software engineering concepts and tools.
CO3	Develop the software design to meet customer expectations using modelling language.
CO4	Identify and use various cost estimation techniques used in software engineering project management.
CO5	Apply verification and validation techniques on a given software project.
CO6	Demonstrate deployment and basic maintenance skills.

Course Contents:

Unit I- Introduction to software engineering: Code comprehension.

Unit II- Requirement engineering: Requirement Elicitation, specification, IEEE standard template for SRS, Requirement Engineering tools.

Unit III- UML Modeling: Use case diagram , State diagram, Activity Diagram, Class Diagram, Sequence diagram, Collaboration diagram, Deployment Diagram, Component Diagram, Event trace diagram , c++ code generation, Introduction to Sec UML.

Unit IV- Software Metrics: Product, process and project metrics.

Unit V- Software Testing: Structural testing, functional Testing, Testing Strategies and Tactics, Test Case design.

Note: Lab exercises will be based on project-oriented. Each student has work on a project and completes the development of software product using software engineering principles

Evaluation Scheme:

Exams		**Marks**	**Coverage**
P-1		15 Marks	**Software Analysis and design**
P-2		15 Marks	**Software implementation and testing**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Text Books

1. Software Engineering: A practitioner's approach: Roger S. Pressman, McGraw-Hill Publications (Sixth Edition)
2. The Unified Modeling Language Users Guide: Grady Booch, James Rambaugh, Ivar Jacobson, Addision Wesley

References

1. Modern Structured Analysis: Edward Yourdon , PHI Publications

Web References:

1. https://onlinecourses.nptel.ac.in/noc20_cs68/preview
2. https://online.visual-paradigm.com/
3. https://www.coursera.org/learn/introduction-to-software-engineering

Title of Course: Advanced Programming Lab-3 **Course Code: 18B17CI673**

L-T-P scheme: 0-0-2 **Credit: 3**

Prerequisite: Students must have already registered for the course, "***Introduction to Computers and Programming***" and "**Object Oriented Programming**".

Objective:

1. To learn and be able to implement the front-end and back-end web-technologies.
2. To develop the abilities to call oneself full-stack web developer.

Learning Outcomes:

Course Outcome	**Description**
CO1	Get familiar with processes of Full Stack Web Development.
CO2	Have a good grounding of Web Application Terminologies, Internet tools and languages like HTML5 and CSS.
CO3	Possess demonstrative skills in using and applying JavaScript.
CO4	Build modern, fast and scalable server-side web applications with NodeJS and databases like SQL or MongoDB.
CO5	Apply web engineering approaches required to create web applications
CO6	Work as a team on a project.

Course Content:

Part-1: Fundamentals of Full Stack Web Development

Unit-1 Creating first web-application, hosting a web application, creating websites, authoring tools, domain names.

Part-2: Front End Tools & Technologies

Unit-2 Markup and Styling: HTML, Cascading Style Sheets, using Bootstrap.

Unit-3 JavaScript Fundamentals: Language Features, JSON, Ajax, jQuery, Popular Frameworks like React, Angular JS.

Part-3: Back End Tools & Technologies

Unit-4 Web Programming through Node.js and/or Java. Node.js Modules, NPM, Events, Upload File, Email, Get/Post methods, Java Servlets vs. JSP, Request vs. Response objects, other Java objects and features.

Unit-5 Databases and Web Storage: Designing and creating databases, database connection through back end programming languages, Web storage to store sessions, cookies, and cached data in the browser.

Part-4: Project Development

Unit-6 Using Git, Common Git commands, Project Development.

Teaching Methodology:

This course is introduced to help students transition from a simple developer to a full stack developer. Starting from frontend development, the student will slowly progress to become to other aspects of development including backend, database, version control and other essential technologies that are helpful for a developer. The entire course is broken down into four separate parts: Fundamentals of Web Development, Front End tools & Technologies, Back End Tools & Technologies, and Project Development. Each section includes multiple technologies to help a student gain more experience as a developer. This lab course is well complemented by a theory course under the name Web Development in the same semester that helps a student learn and discuss the technical details of the underlying technologies.

Evaluation Scheme:

Exams		**Marks**	**Coverage**
P-1		15 Marks	**Based on Lab Exercises: 1-7**
P-2		15 Marks	**Based on Lab Exercises: 8-14**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Web Technology Lab (will be added from time to time): Digital copy will be available on the JUET server.