DEPARTMENT OF COMPUTER ENGINEERING
B.TECH. II YEAR (4YDC)
SEMESTER-A
CO 24057: OBJECT ORIENTED PROGRAMMING SYSTEMS

Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	1	2	4	1	70	30	40	60	200

PRE-REQUISITES: CO10504: Computer Programming

COURSE OBJECTIVES: The objective of course is to develop programming skills of students, using object oriented programming concepts, learn the concept of class and object using Java and develop real world applications.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Explain various concepts of object oriented terminology.
2. Define and implement the concepts of data encapsulation, abstraction, inheritance and polymorphism.
3. Design and execute quality programs using exception handling.
4. Solve the real world business problems as per specifications.

COURSE CONTENTS:
THEORY:

UNIT 1 Introduction to Object Oriented Thinking & Object Oriented Programming: Comparison with Procedural Programming, features of Object oriented paradigm, Merits and demerits of OO methodology; Object model; Elements of OOPS, IO processing.

UNIT 2 Encapsulation and Data Abstraction- Concept of Objects: State, Behavior & Identity of an object; Classes: identifying classes and candidates for Classes Attributes and Services, Access modifiers, Static members of a Class, Instances, Message passing, and Construction and destruction of Objects.

UNIT 3 Relationships – Inheritance: purpose and its types, 'is a' relationship; Association, Aggregation. Concept of interfaces and Abstract classes.

UNIT 4 Polymorphism: Introduction, Method Overriding & Overloading, static and run time Polymorphism.

UNIT 5 Strings, Exceptional handling, Introduction of Multi-threading and Data collections. Case study like: ATM, Library management system.

DIRECT ASSESMENT:
ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

ASSESSMENT OF PRACTICAL:

1. Internal Assessment for continuous evaluation (40%): Lab assignments, demonstration, Viva, File etc.
2. End semester Practical Exam (60%): Quiz/Programming test, lab journal, demo, viva etc.

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Timothy Budd, "An Introduction to Object-Oriented Programming", Addison-Wesley Publication, 3rd Edition 2002.
2. Cay S. Horstmann and Gary Cornell, "Core Java: Volume I, Fundamentals", Prentice Hall publication, 2007.

REFERENCE BOOKS:

1. G. Booch, "Object Oriented Analysis & Design", Addison Wesley, 2006
2. James Martin, "Principles of Object Oriented Analysis and Design", Prentice Hall/PTR, 1992.
3. Peter Coad and Edward Yourdon, "Object Oriented Design", Prentice Hall/PTR, 1991.
4. Herbert Schildt, "Java 2: The Complete Reference", McGraw-Hill Osborne Media, 11th Edition, 2018.

DEPARTMENT OF COMPUTER ENGINEERING
B.TECH. II YEAR (4YDC)
SEMESTER-A
CO 24009: COMPUTER ARCHITECTURE

Hoursper Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	2	3	1	70	30	40	60	200

PRE-REQUISITES: NIL

COURSE OBJECTIVES:
The main objective of this course is to compare various architectures of Computers and their components like memory etc. and to develop the skills of the students to write the assembly language programs for various instructions.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Illustrate architecture of a computer, its components and their interconnection.
2. Describe execution of instruction in a computer.
3. Identify the addressing modes used in macro instruction.
4. Design programs in assembly language and justify the importance of parallel architecture.

COURSE CONTENTS:
THEORY:

UNIT 1 Introduction, Milestones in Computer Architecture, Von Neumann Model: Processor Organization- ALU, Control Unit; System Bus, Memory, I/O Devices. Multilevel model of Computer, Concept of Instruction Execution.

UNIT 2 Review of Combinational and Sequential Circuits. Memory Organization: Memory Hierarchy, Memory Properties, Main Memory, Associative Memory, Cache Memory. Machine Language Level (ISA level): Instruction Formats, Addressing Modes, Instruction Types, Flow of Control. RISC v/s CISC.

UNIT 3 Memory mapped I/O and I/O mapped I/O, I/O Techniques: Programmed I/O, Concept of Interrupts, Interrupt driven I/O and DMA, I/O Device Interfaces, I/O Processors. Serial and Parallel Communication. Computer Buses.

UNIT 4 Concept of Hardwired and Micro Programmed Control. Micro Instructions, Instruction Fetch and Queuing, Micro Instruction Control, Design of the Micro Architecture Level.

UNIT 5 Parallel Architectures: On-chip Parallelism- Instruction Level Parallelism, On-chip Multithreading, Multicore Processor Architecture. Pipelining: RISC Pipeline, Exception handling of Pipelining, Hazards of Pipelining.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

ASSESSMENT OF PRACTICAL:

1. Internal Assessment for continuous evaluation (40%): Lab assignments, demonstration, Viva, File etc.
2. End semester Practical Exam (60%): Quiz/Programming test, lab journal, demo, viva etc.

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. William Stallings, “Computer Organization and architecture”, Ninth Edition, Pearson, 2012.
2. Tannenbaum and Austin, “Structured Computer Organization”, Sixth edition, PHI, 2013.
3. Michael J. Flynn “Computer Architecture: Pipelined and Parallel Processor Design, First Edition, 1995.

REFERENCE BOOKS:

1. V. Carl Hamacher, “Computer Organization”, Fifth Edition, 2011, McGraw Hill.
2. John P. Hayes, “Computer Architecture and Organization”, Fourth Edition, TMH, 2003
3. Morris Mano, “Computer System Architecture”, Third Edition, 2007, PHI.
4. David A. Patterson and John L. Hennessy, “Computer Organization and Design: The Hardware/Software Interface”, Fourth Edition, Morgan Kauffman, 2011

DEPARTMENT OF COMPUTER ENGINEERING
B.TECH. II YEAR (4YDC)
SEMESTER-A
CO 24997: PROGRAMMING PRACTICES

Hoursper Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
-	1	2	-	2	-	-	40	60	100

PRE-REQUISITES: CO10504: Computer Programming

COURSE OBJECTIVES: To give hands-on experience to students on different components of good programming.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Develop programs for complex real world problems.
2. Apply good programming practices in their code like Comments, indentation etc.
3. Utilize Debugger and its tools like gdb/gnu for error handling.
4. Demonstrate configuration and usage of different software tools used in industry.

COURSE CONTENTS:

UNIT 1 Getting Familiar With Linux (Ubuntu/CentOS)

Lab 1. Doing regular activities:
- (a) Changing wallpaper.
- (b) Playing multimedia files (video and audio).
- (c) Creating documents (docs, ppts and spreadsheets).
- (d) Install softwares through repositories and different commands.
- (e) Editing images using GIMP (MS Paint like software).
- (f) Text Editors (Vi/Vim, Gedit, Emacs, Sublime Text etc.)

Lab 2. Bash commands and Linux utilities:
- (a) Find files on hard disk using locate and find.
- (b) Find within a file (text or pdf).
- (c) sed and awk.
- (d) View current IP address of my computer/workstation and how to change it.
- (e) Rename a file or folder.
- (f) Move/copy file/folder from one place to another.
- (g) Change password of your user account.
- (h) View hardware specification of my computer/workstation (CPU Model, RAM, HDD etc.).
- (i) Documentation of commands (man, help, info).

UNIT 2 Programming

Lab 3. Introduction to file handling, structures and pointers in C.

Lab 4. Introduction to Python 3.
- (a) Inbuilt data types (list, dictionary, sets etc.)
- (b) Control structures (e.g. if -else, loops, functions)
- (c) Introduction to Jupyter notebook.

Lab 5. Coding style.
(a) GNU Coding Standards
(b) Writing meaningful comments.
(c) Appropriate variable/function/ le names.
(d) Indentation.

UNIT 3 Debugging using GDB.
Lab 6 (a) Listing source code lines.
(b) Compiling with debugging enabled.
(c) Starting a program step by step.
(d) Execute one step at a time, one function at a time, stepping out of a function.
(e) Printing values of variables, structures, arrays, strings, pointers, linked lists etc.
(f) Examining addresses (x).
(g) Stack frames.
Lab 7 (j) Running a program till the first breakpoint or run time error.
(k) Listing local, global variables. (using info, locals, all variables)
(l) Watches (list watches[info watch], write watches, read watches)
(m) Breakpoints (on line numbers, functions etc.)
(n) Continue (till next breakpoint or watch).
(o) Current line address (frame, where).
(p) Code pro ling.

UNIT 4 Project management
Lab 8. (a) Subversion control using Git and GitHub/BitBucket.
(b) Using Ant.
(c) Download and modify code of an open source software.
Lab 9 Using an IDE.
(i) Create and manage projects.
(ii) Debug in CodeBlocks/IDE.
(iii) Execute/build fies/projects.
(iv) Auto indent, bulk comments.
(v) Auto complete identifier names.

UNIT 5 Scientific Documentation
Lab 10 LaTeX and Overleaf.com.
(a) Text formatting (bold, italic, size, colour etc.)
(b) Types of scientific documents (research papers, reports etc.)
(c) Setting margins
(d) itemize, enumerate.
(e) Mathematical Symbols (e.g. ; ; ac++db ; P; Q; 1; px + y + z, etc.)
(f) Equations.
(g) Sections, subsections, subsubsections ...
(h) Figures and tables.
(i) referencing sections, figures, tables, equations, references, etc.
(j) BibTex and references.

DIRECT ASSESMENT:

ASSESSMENT OF PRACTICAL:

1. Internal Assessment for continuous evaluation (40%): Lab assignments, demonstration, Viva, File etc.
2. End semester Practical Exam (60%): Quiz/Programming test, lab journal, demo, viva etc.

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. "Learn Python 3 The Hard Way" by Zed A. Shaw, Addison-Wesley Publisher, January 2017.
2. "The Art of Readable Code" by Dustin Boswell, Trevor Foucher, O'Reilly Publication, Inc. November 2011.
3. "Art of Debugging with GDB, DDD, and Eclipse" by Norman Matloff and Peter Jay Salzman, No Starch Press, September 2008.
4. "LaTeX Cookbook" by Stefan Kottwitz, Packt Publishing, October 2015.
5. "Learning IPython for Interactive Computing and Data Visualization" by CyrilleRossant, 2nd Edition, Pakt Publication, October 2015.
6. "Mastering Git" By Jakub Narębski, Pakt Publication, April 2016.

REFERENCE BOOKS:

1. GNU GDB Documentation, https://www.gnu.org/software/gdb/documentation/
2. TeX User Group, https://www.tug.org/
3. Google CoLab for Jupyter, https://research.google.com/colaboratory/