



**B. Tech.
Semester VI**

**MORDERN PROGRAMMING
TECHNOLOGIES**

XXXXXX

EFFECTIVE FROM July-2021

Syllabus version: 1.00

Subject Code	Subject Title	Teaching Scheme			
		Hours		Credits	
		Theory	Practical	Theory	Practical
XXXXXX	Modern Programming Technologies	3	2	3	1

Subject Code	Subject Title	Theory Examination Marks		Practical Examination Marks	Total Marks
		Internal	External	CIE	
XXXXXX	Modern Programming Technologies	40	60	50	150

Objectives of the course:

- To explain core python programming concepts including data types, object-oriented programming, file handling, and exception handling.
- To illustrate applications of user-defined data structures, modules, networking, and system programming.
- To inculcate skills in students to develop GUI based application.

Course outcomes:

Upon completion of the course, the student shall be able to

C01: Learn features, basic syntax and process of Rust variables, constant, data types and function.

C02: Apply concepts of structure and method to solve computational problems.

C03: Indicate usage of object-oriented programming and various modules.

C04: Understand Go basics, including data types, variables, and control structures.

C05: Explore complex types, such as slices, functions, structs, and interfaces.

C06: Learn how to create your own package and run programs concurrently.

Sr. No.	Topics	Hours
Unit – I		
1	Rust: Common Programming Concept: Variables and mutability - Differences between variables and constants, Shadowing; Data Types - Scalar types, Compound types; Functions - Function parameters, statements, Expressions in function bodies, Functions with return values; Comments, Control flow - if expressions, Repetition with loops.	8

	Rust: Understanding Ownership: What is ownership - Ownership rules, Variable scope, The string type, Memory and allocation, Ownership and functions, Return values and scope; References and borrowing - Mutable references, dangling references, The rules of references; The slice type - String slices, Other slices.	
Unit – II		
2	Rust: Using Structs to Structure related Data: Defining and instantiating Structs - Using the field init shorthand when variables and fields, Have the same name, Creating instances from other instances with struct update syntax, Using tuple structs without named fields to create different types, Unit-like structs without any fields; An example program using structs - Refactoring with tuples, Refactoring with structs - Adding more meaning, Adding useful functionality with derived traits; Method syntax - Defining methods, methods with more parameters, Associated functions, Multiple impl blocks.	7
Unit – III		
3	Rust: ENUMS and Pattern Matching: Defining an Enum - Enum values, The option enum and its advantages over null values; The match control flow operator - Patterns that bind to values, Matching with option<T>, Matches are exhaustive, The _ placeholder; Concise control flow with if let. Rust: Managing Growing Projects with Packages Crats and Modules: Packages and crates, Defining modules to control scope and privacy, Paths for referring to an item in the module tree - Exposing paths with the pub keyword, Starting relative paths with super, Making structs and enum public; Bringing paths into scope with the use keyword - Creating idiomatic use paths, Providing new, Names with the as keyword, Re-exporting names with pub use, Using external packages, Using nested paths to clean up large use lists, The glob operator, Separating modules into different files.	8
Unit – IV		
4	Go: Introduction: Machine setup, First program, Read a go program. Types: Number, Strings, Booleans. Variables: Name a variable, Scope, Constants, Multiple Variable. Control Structures: For statement, If statement, Switch statement.	8

Unit – V		
5	Go: Array, Slices and Maps: Arrays, slices: append, copy, maps	7
	Functions: Second functions, Variadic functions, Closure, Recursion, Defer panic and recover, Pointers.	
	Structs and Interfaces: Structs, Methods, Interfaces.	
Unit – VI		
6	Go: Packages and Testing: The core packages, Severs, Creating packages, Testing.	7
	Concurrency: Go routines, Channels.	

Sr. No.	Modern Programming Technologies (Practical)	Hours
1	Explore control flow and looping.	2
2	Explore slice type.	2
3	Explore referencing and borrowing.	4
4	Explore enum.	4
5	Explore Packages and crats.	4
6	Explore keywords and globe operator.	2
7	Explore different variable types, scope and constant.	2
8	Explore array, slice and map. Explore different kind of methods. Explore structs and interface.	6
9	Explore package and concurrent.	4

Text book:

1. Steve Klabnik, Carol Nicholas – “The Rust Programming Language”, No Starch Press.
2. Caleb Doxsey – “Introducing Go Build Reliable, Scalable Programs”, O’Reilly Publication.

Reference books:

1. Jim Blandy, Jason Orendorff - “Programming Rust – Fast, Safe System Development”, O’Reilly Publication.
2. Alan A. A. Donovan, Brian W. Kernighan - "The Go Programming Language", Addison-Wesley.

Course objectives and Course outcomes mapping:

- To explain core python programming concepts including data types, object-oriented programming, file handling, and exception handling: C01, C02, C03, C04
- To illustrate applications of user-defined data structures, modules, networking, and system programming: C03, C06
- To inculcate skills in students to develop GUI based application: C05

Course units and Course outcomes mapping:

Unit No.	Unit Name	Course Outcomes					
		C01	C02	C03	C04	C05	C06
1	Rust: Common Programming Concept and Understanding Ownership	✓					
2	Rust: Using Structs to Structure related Data		✓				
3	Rust: ENUMS and Pattern Matching and Managing Growing Projects with Packages Crats and Modules			✓			
4	Go: Introduction, Types, Variables and Control Structures				✓		
5	Go: Array, Slices and Maps, Functions, Structs and Interfaces:					✓	
6	Go: Packages and Testing, Concurrency						✓

Programme outcomes:

- PO 1: Engineering knowledge: An ability to apply knowledge of mathematics, science, and engineering.
- PO 2: Problem analysis: An ability to identify, formulates, and solves engineering problems.
- PO 3: Design/development of solutions: An ability to design a system, component, or process to meet desired needs within realistic constraints.
- PO 4: Conduct investigations of complex problems: An ability to use the techniques, skills, and modern engineering tools necessary for solving engineering problems.
- PO 5: Modern tool usage: The broad education and understanding of new engineering techniques necessary to solve engineering problems.
- PO 6: The engineer and society: Achieve professional success with an understanding and appreciation of ethical behaviour, social responsibility, and diversity, both as individuals and in team environments.
- PO 7: Environment and sustainability: Articulate a comprehensive world view that integrates diverse approaches to sustainability.
- PO 8: Ethics: Identify and demonstrate knowledge of ethical values in non-classroom activities, such as service learning, internships, and field work.

- PO 9: Individual and team work: An ability to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give/receive clear instructions.
- PO 11: Project management and finance: An ability to demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- PO 12: Life-long learning: A recognition of the need for, and an ability to engage in life-long learning.

Programme outcomes and Course outcomes mapping:

Programme Outcomes	Course Outcomes					
	C01	C02	C03	C04	C05	C06
P01						
P02		✓	✓		✓	✓
P03						
P04						
P05						
P06						
P07						
P08						
P09	✓	✓	✓	✓	✓	✓
P010						
P011						
P012						