

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT (AN AUTONOMOUS INSTITUTION) (Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad) Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act, 1956

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2430598 PARALLEL COMPUTATION - RUST LAB

B. Tech. II Year-II Sem

L / T / P / C 0 / 0 / 2 / 1

COURSE OUTCOMES - CO'S

- Identify the key features and applications of Rust in areas like systems programming, embedded development, and web applications.
- Explain and implement Rust's ownership, borrowing, and lifetime rules to manage memory safely and efficiently.
- Create reusable functions with well-defined ownership behavior.
- Organize projects using packages, crates, and modules for scalability and modularity.
- Understand and apply Rust's object-oriented features such as traits, trait objects, and design patterns for polymorphism.

LIST OF EXPERIMENTS:

Module-I:

Key features of Rust, Applications of Rust (systems programming, embedded systems, web development), Download and install Rust compiler (rustc) and package manager (cargo),Setting up an Integrated Development Environment (IDE) with Rust support

Program structure (main function, modules), Variables and Data Types (int, float, char, bool, etc.) ,Operators (Arithmetic, Relational, Logical, Assignment)

1. Hello, World!

- Basic Rust program structure using fn main().
- 2. Variable Declaration and Mutability
 - Using let, mut, and type inference.

3. Data Types Demo

• Use i32, f64, char, bool, etc., in a single program.

4. Arithmetic and Logical Operators

• Calculator-style program showing arithmetic and logical operations.

5. Using cargo for a Simple Project

 \circ $\,$ Create a simple project using cargo new, build, and run it.



MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT (AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad) Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act, 1956

Module II:

Understanding ownership model (memory management in Rust), Taking ownership (moving) vs. borrowing data with references (&), Mutable vs. immutable borrows, Lifetime annotations

Conditional statements (if, else if, else), Looping statements (for, while, loop), Pattern matching (powerful conditional branching in Rust), Nested statements.

6. Ownership Transfer

• Pass variables to functions and explore ownership and move.

7. Borrowing and References

- \circ $\;$ Demonstrate &T and &mut T references and restrictions.
- 8. Lifetimes Example
 - \circ $\;$ Use lifetime annotations in a function returning references.

9. If-Else Statement

 \circ Even/odd checker or grade calculator.

10. Looping Examples

 \circ $\;$ Use for, while, and loop to iterate over arrays or counters.

11. Pattern Matching with match

 \circ $\;$ Match numbers to print weekdays or match enums.

12. Nested Control Statements

• Combine loops and conditionals in one program (e.g., a simple number guessing game).

Module III:

Defining and calling functions, Function arguments and return values, Ownership and borrowing rules within functions

Structuring Data: Using Structs, Initializing Structs - Field Init Shorthand, Tuple Structs, Examples of Struct - Area of Rectangle, Enum and Pattern Matching: Enums, Option Enum, Match statement and Enum

13. Functions with Arguments and Return Values

• Create a function to calculate factorial or square of a number.

14. Ownership in Functions

• Pass by value vs. reference in function parameters.

15. Using Structs

• Define a Rectangle struct and calculate area.

16. Tuple Structs and Field Init Shorthand

 \circ Define a Color(u8, u8, u8) and use field shorthand for init.

$17.\ensuremath{\,\text{Enums}}$ and Pattern Matching

 \circ $\;$ Define an enum for TrafficLight and match its values.



MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT (AN AUTONOMOUS INSTITUTION)

(An AUTONOMOUS INSTITUTION) (Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad) Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act, 1956

• Safe division function returning Option<f64>.

Module IV:

Managing Growing Projects with Packages, Creates and Modules

Error handling in Rust, Unrecoverable errors with panic! Recoverable errors with Result enum, Shortcuts to Panic with Error: unwrap and expect

Smart Pointers: Smart Pointers in Rust, Box<T> Smart Pointer, Deref trait, Drop trait, Rc<T> Smart Pointer, RefCell<T>

19. Error Handling with Result

• File reading or division with error handling using Result.

20. Using Box, Rc, and RefCell

• Demonstrate smart pointers with a simple linked list or counter program.

Module V:

Concurrency: Threads, Shared state concurrency, Extensible concurrency

Object-oriented Programming features of RUST: Characteristics of Object-Oriented Languages, Using Trait Objects That Allow for Values of Different Types, Implementing an Object-Oriented Design Pattern

21. "Polymorphic Behavior with Trait Objects: A Speakable Animal Zoo"

Covers: Traits, dynamic dispatch, trait objects (&dyn Trait)

22. "Implementing Strategy Pattern with Traits for Payment Processing" Covers: Object-oriented design using traits and Box<dyn Trait> for dynamic strategy switching

23. "**Trait-Based Drawing Application with Heterogeneous UI Components**" *Covers:* Object-oriented characteristics, trait objects, and allowing different drawable types in a single collection



MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT (AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad) Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act, 1956