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

ONE WEEK NATIONAL LEVEL FDP REPORT

ON

RUST PROGRAMMING

Title: FDP on Rust programming

Program Date: 23/6/2025to 28/6/2025

Venue: Hamsasana computer Center, MG010

Speaker: Mr. Rajesh Kumar, Technical Trainer (Corporate Associates & college Grads)

Organized by: Department of CSE (Data Science), MLRITM

FDP Objectives:

- To familiarize participants with the core concepts and syntax of Rust.
- To understand Rust's scalability, concurrency, and efficiency for building modern applications.
- To explore how Rust supports IoT (Internet of Things) and system-level programming.
- To demonstrate how older Rust code remains compatible with new versions, ensuring long-term maintainability.

Major Outcomes

- **Comprehensive Understanding of Rust Language:**
  Participants gained a solid foundation in Rust programming, including ownership, borrowing, and concurrency concepts.
- **Hands-on Experience:**
  Faculty members and professionals practiced coding exercises, exploring real-time examples in system-level and IoT programming using Rust.
- **Enhanced Technical Skills:**
  Attendees improved their skills in memory management, performance optimization, and safe concurrency handling—key strengths of Rust.
- **Application-Oriented Learning:**
  The FDP emphasized developing scalable, concurrent, and efficient applications suitable for modern IT and industrial environments.
- **Exposure to Industry Trends:**
  Through expert sessions, participants learned how Rust is used in **Data Engineering**, **Business Intelligence**, and **High-Performance Computing** projects.
- **Encouragement for Research and Innovation:**
  The programme inspired faculty to pursue research in system-level programming, embedded systems, and next-generation software architectures using Rust.
- **Collaborative Learning Environment:**
  Networking among participants, researchers, and industry trainers fostered collaboration and idea exchange for future academic and industrial projects.
- **Curriculum Enrichment:**
  Faculty participants received insights to integrate Rust programming concepts into their courses and laboratory practices.

- **Professional Development Certification:**
  All participants received certificates acknowledging their completion of the FDP, enhancing their professional profile.

FDP RUST Course Content Training Schedule

S. No	Date	Module	Topic
1	**23-Jun-25**	Basics of Rust	What is Rust? Why Rust? Use case of Rust Basic Rust script for Hello World. Variable Declaration and Mutability Data Types Demo Operators and Types of Operators Arithmetic and Logical Operators
2	**24-Jun-25**	Basics of Rust	Using cargo for a Simple Project - How to install Cargo Tool and Configure for Sample Rust Project
3	**24-Jun-25**	Ownership and Control Flow	Ownership Transfer Borrowing and References Lifetimes Example
4	**25-Jun-25**	Ownership and Control Flow	If-Else Statement Looping Examples Pattern Matching with match Nested Control Statements
5	**26-Jun-25**	Functions and structs	Functions with Arguments and Return Values Ownership in Functions Using Structs Tuple Structs and Field Init Shorthand Enums and Pattern Matching Option Enum Usage
6	**27-Jun-25**	Project Management and Smart Pointers Object oriented Programming	Error Handling with Result Using Box, Rc, and RefCell
7	**28-Jun-25**	Object oriented Programming	"Polymorphic Behaviour with Trait Objects: A Separable Animal Zoo "Implementing Strategy Pattern with Traits for Payment Processing" "Trait-Based Drawing Application with Heterogeneous UI Components"

DAY 1: 23-06-2025

Module: ***Basics of Rust***

Topics Covered:

- Introduction to Rust: *What is Rust? Why Rust?*
- Use cases of Rust in system programming, web development, and embedded systems.
- Writing the first Rust program: *Hello, World!*
- Variable declaration, mutability, and scope rules.
- Understanding Rust's static typing system and data types.
- Demonstration of arithmetic and logical operators.

Session Description:
Day 1 introduces participants to the fundamentals of the Rust programming language. The session explains why Rust has gained prominence for safe and efficient system-level programming. Participants write their first Rust program and explore key features such as mutability and type safety. Through hands-on examples, they learn how Rust's strong typing and ownership model enhance reliability and performance.

Expected Outcomes:

- Understand the motivation behind using Rust.
- Gain familiarity with basic syntax and Rust tools.
- Write and execute simple Rust programs.
- Learn how data types and mutability work in Rust.

DAY 2: 24-06-2025

Morning Session: Basics of Rust (continued)

Topics Covered:

- Introduction to *Cargo* – Rust's build system and package manager.
- Installing and configuring Cargo.
- Creating and managing a simple Rust project using Cargo.

Session Description:
Participants learn about Cargo, an essential tool for building, running, and managing Rust projects. The session focuses on setting up a project structure, adding dependencies, and compiling code efficiently using Cargo commands.

Expected Outcomes:

- Install and configure Cargo successfully.
- Create, build, and run Rust projects with Cargo.
- Understand dependency management using *Cargo.toml*.

Afternoon Session: Ownership and Control Flow (Part 1)

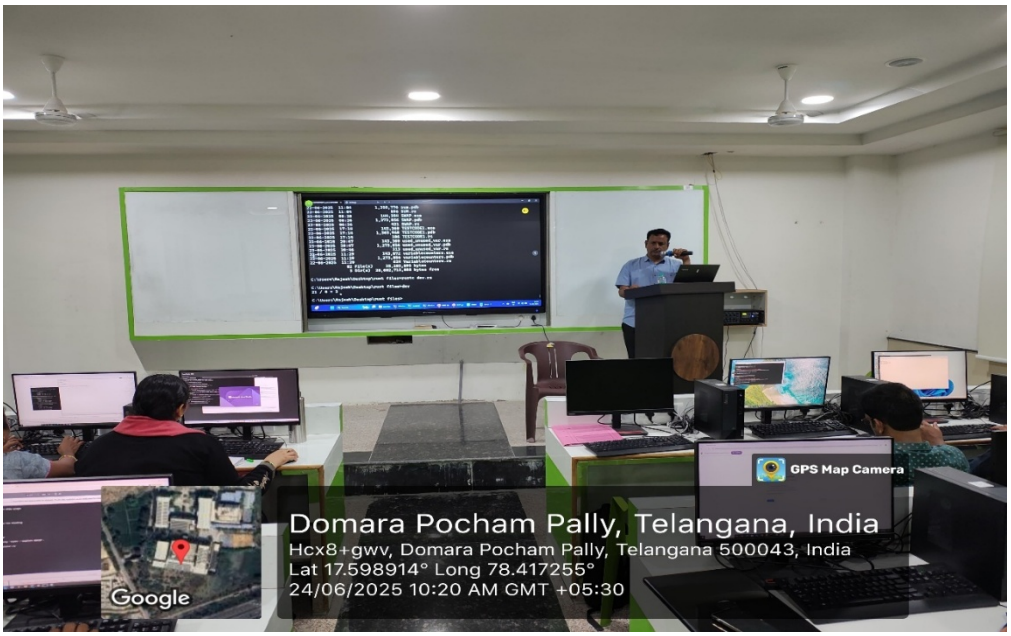
Topics Covered:

- Understanding Rust's *Ownership Model*.
- Ownership transfer and memory safety.
- Borrowing and references.
- Introduction to lifetimes and their importance.

Session Description:
This session focuses on Rust's unique memory management approach without garbage collection. Through practical coding exercises, participants grasp how ownership, borrowing, and lifetimes ensure memory safety and concurrency without data races.

Expected Outcomes:

- Understand the ownership model and how Rust ensures memory safety.
- Use references and borrowing correctly.
- Recognize the role of lifetimes in avoiding dangling references.



DAY 3: 25-06-2025

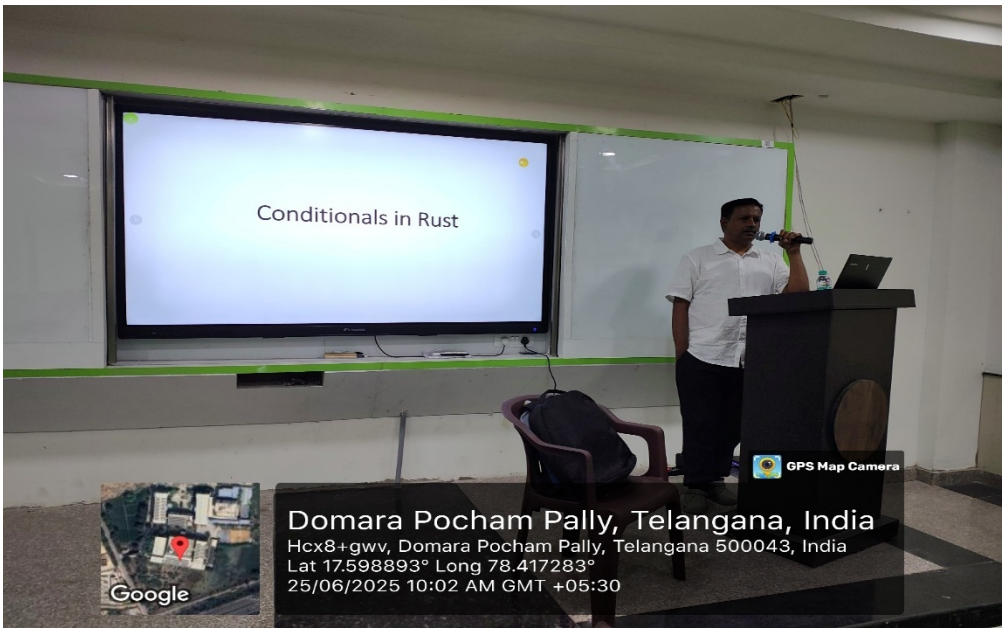
Module: ***Ownership and Control Flow (continued)***

- **Topics Covered:**
  - Conditional control using *if-else* statements.
  - Implementing loops (`loop`, `while`, and `for`).
  - Pattern matching using the `match` statement.
  - Nested control structures.

- **Session Description:**

Day 3 emphasizes Rust's control flow mechanisms and how pattern matching simplifies complex branching logic. Participants implement examples demonstrating nested control structures and decision-making in Rust.

- **Expected Outcomes:**
  - Apply conditional statements and loops effectively.
  - Utilize pattern matching for clean and readable code.
  - Gain confidence in handling control flow within Rust programs.



DAY 4: 26-06-2025

Module: ***Functions and Structs***

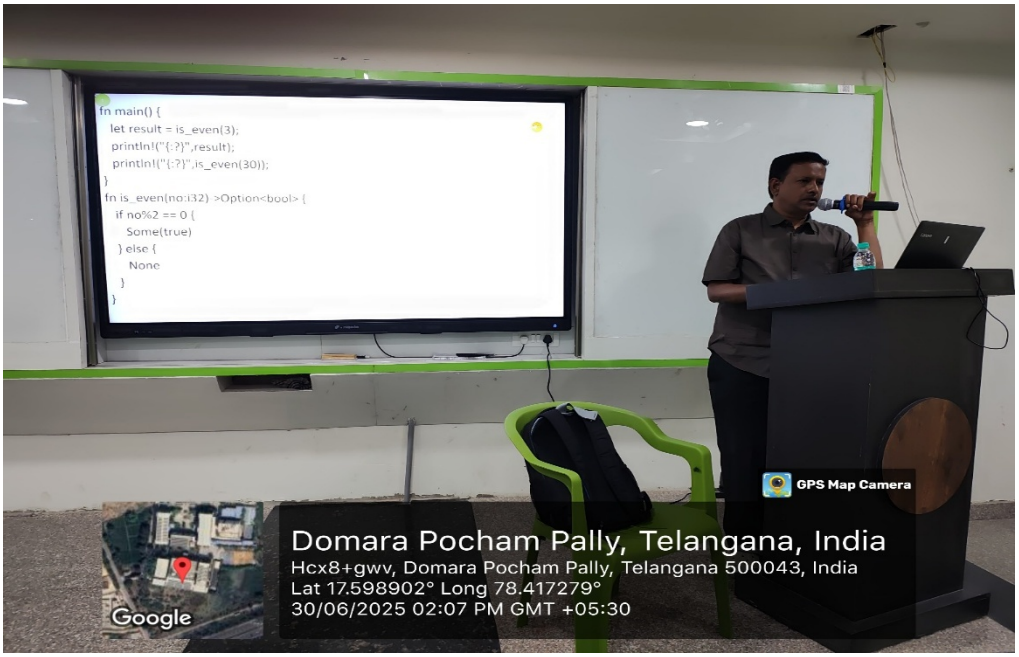
- **Topics Covered:**
  - Writing and using functions with parameters and return values.
  - Ownership and borrowing within functions.
  - Introduction to structs and their role in data organization.
  - Tuple structs and field initialization shorthand.
  - Defining and using enums with pattern matching.
  - Understanding the `Option` enum for handling nullable values.

- **Session Description:**

This session explores modular programming using functions and data abstraction using structs and enums. Participants design reusable functions, understand data encapsulation, and implement enums to represent multiple states of data.

- **Expected Outcomes:**
  - Write modular code using functions.
  - Design custom data types using structs and enums.

- Implement safe null handling using the `Option` enum.
- Understand ownership implications within functions.



DAY-5:27-06-2025.

Module: ***Project Management and Smart Pointers***

- **Topics Covered:**
  - Introduction to Object-Oriented Programming (OOP) concepts in Rust.
  - Error handling using the `Result` type.
  - Understanding and using smart pointers: `Box`, `Rc`, and `RefCell`.

- **Session Description:**

Day 5 bridges traditional OOP concepts with Rust's trait-based design. Participants learn how smart pointers and ownership interact to manage heap-allocated data. Practical demonstrations show how `Result` is used for robust error handling.

- **Expected Outcomes:**
  - Manage heap data using smart pointers.
  - Apply Rust's error-handling patterns effectively.
  - Implement OOP-like structures using traits and data encapsulation.

DAY-6:28-06-2025.

Module: ***Advanced Object-Oriented Concepts with Traits***

- **Topics Covered:**
  - Implementing polymorphism using trait objects.
  - Case study: *A Separable Animal Zoo* – Demonstrating dynamic dispatch.
  - Implementing the *Strategy Pattern* using traits for payment processing systems.
  - Trait-based drawing application with heterogeneous UI components.

- **Session Description:**

The final day focuses on building complete object-oriented solutions using Rust traits. Participants explore trait-based polymorphism and learn to design extensible applications. Real-world examples demonstrate how traits replace traditional class inheritance models.

- **Expected Outcomes:**
  - Design and use trait objects for polymorphic behavior.
  - Implement design patterns such as Strategy in Rust.
  - Build trait-based modular applications combining multiple object types.
  - Gain confidence to apply Rust in real-world software projects.

Conclusion

The **Faculty Development Programme on Rust Programming**, conducted from **23rd to 28th June 2025**, successfully provided an in-depth exploration of modern system programming concepts using Rust. Over six days, participants gained both theoretical understanding and practical experience in writing efficient, safe, and concurrent Rust applications. The FDP began with the **fundamentals of Rust**, including syntax, variables, data types, and operators, giving participants a strong foundation to appreciate the language's design philosophy of safety and performance. The sessions on **Cargo project management** equipped attendees with essential skills to organize, build, and manage Rust projects effectively. Through the detailed modules on **ownership, borrowing, and lifetimes**, participants developed a clear understanding of Rust's unique approach to memory management—an essential concept that sets Rust apart from traditional programming languages. The exploration of **control flow mechanisms, pattern matching, functions, and structs** further enhanced their programming proficiency and understanding of data abstraction. In the later stages, participants were introduced to **advanced topics such as smart pointers, error handling, and object-oriented design using traits**, which demonstrated Rust's flexibility and modern design patterns. Hands-on exercises, including *trait-based applications* and *strategy pattern implementations*, reinforced practical problem-solving skills. The FDP successfully met its objective of **empowering educators and developers with modern programming practices** aligned with industry trends. Participants expressed high levels of engagement and satisfaction, recognizing Rust as a promising language for building secure, high-performance applications. This programme not only strengthened the faculty's programming expertise but also encouraged interdisciplinary collaboration, innovation, and research in **safe systems programming and emerging technologies** using Rust.

MARRI LAXMAN REDDY
Institute of Technology and Management
(Approved By AICTE, New Delhi & Affillated to JNTU Hyderabad)
Dundigal,Medchal(M),Hyderabad,Telangana, India – 500 043.
(UGC - AUTONOMOUS)

List of Participants for the F D P R U S T P R O G R A M M I N G

SNO	NAME OF THE FACULTY	FACULTY ID	DEPARTMENT
1	Dr. Y. MADHUSEKHAR	MLRS10059	CSE
2	Dr PARVES MOHAMMED	MLRS10040	CSE
3	Dr. B. PRAVEEN	MLRS10086	IT
4	Dr. T.S. SRINIVAS	MLRS10162	CSE
5	Dr. SHAIK MAHEBOOB	MLRS10041	CSE
6	Dr. B. REBECCA	MLRS10103	CSC
7	Dr. SHAIK ABDUL JAFFAR	MLRS10436	CSD
8	Mrs. ASMA SULTANA	MLRS10078	CSE
9	Ms. SHAIK NAZMA	MLRS10421	IT
10	Mrs. KATTABOYINA ANUSHA	MLRS10449	IT
11	Mrs. V. SUDHA RANI	MLRS10074	CSE
12	Ms. K. JAYASRI (Ph.D)	MLRS10101	CSD
13	Mr. A. SATCHIDANANDAM	MLRS10156	IT
14	Mrs. K. PALLAVI REDDY	MLRS10136	CSE
15	Mrs. NIVEDITHA	MLRS10418	IT
16	Mrs. G. MOUNIKA	MLRS10456	IT
17	Mrs. S. DEVI PARVATHI	MLRS10165	CSD
18	Mr. P. JAGAN MOHAN	MLRS10168	CSD
19	Mrs. CH. SRAVANI	MLRS10034	CSE
20	Mrs. KOLLI ANUSHA	MLRS10037	CSE
21	K. SUNIL KUMAR	MLRS10064	CSE
22	Mr. K. SURESH	MLRS10029	CSE
23	Mr. Y. PRASHANTH	MLRS10123	CSE
24	Mr. K. VIVEK	MLRS10124	CSE
25	Mr. P. RAMESH BABU	MLRS10032	CSE
26	Mr. V. NARESH KUMAR REDDY	MLRS10126	CSE
27	Mrs. KONDAPATI ANUSHA	MLRS10052	CSE
28	Mr. G. NARASAIAH	MLRS10046	CSE
29	Mrs. A. DURGA BHAVANI	MLRS 10390	CSE
30	Mrs. K. SANDHYA	MLRS10455	IT
31	Mrs. K. HARIKA	MLRS10438	IT
32	Mr. J. SRIKANTH	MLRS10087	CSE
33	Mrs. B. SWAPNA	MLRS 10060	CSM
34	Mrs. D. G. DIVYA	MLRS 10143	CSM
35	Mr. D. KAUSHIK	MLRS10435	IT
36	Mr. RAPOLU SATEESH	MLRS10148	CSE
37	Mr. K. SIVA RAMA PRASAD	MLRS10159	CSE
38	Ms. KESTHA RAVALI	MLRS10424	IT
39	Mrs. D SUNITHA	MLRS10066	CSE
40	Mrs. Y. HARITHA	MLRS10150	CSIT
41	Mrs. N. BHARGAVI	MLRS10205	EEE
42	Dr. A. BRAHMA REDDY	MLRS10009	CSE
43	Mrs. A. LEELA SRAVANTHI	MLRS10153	IT
44	Mrs. P. SIVA PADMINI	MLRS10106	CSD