

## NIRMA UNIVERSITY

<b>Institute:</b>	Institute of Technology
<b>Name of Programme:</b>	B.Tech. Computer Science and Engineering
<b>Course Code:</b>	2CSDE91
<b>Course Title:</b>	Contemporary Programming
<b>Course Type:</b>	Departmental Elective
<b>Year of Introduction:</b>	2021-22

### Credit Scheme

L	T	Practical Component				C
		LPW	PW	W	S	
2	0	2	-	-	-	3

### Course Learning Outcomes (CLO):

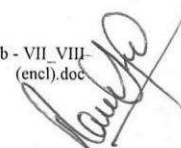
At the end of the course, students will be able to –

1. interpret basic concepts of rust programming language
2. explain the concepts of programming with reference to error handling, message passing and concurrency control
3. develop dynamic programs to solve real-time problems
4. design communication module to create robust programs

### Syllabus:

**Total Teaching hours: 30**

Unit	Syllabus	Teaching hours
Unit-I	<b>Introduction:</b> Introduction to Rust Programming, Installation, History of Rust programming, Why study Rust programming, Environment setup, comments and print statement in Rust. Scalar Data Types, Rust Features	03
Unit-II	<b>Rust structure control and Data:</b> Scalar Types, Compound types, mutability and related concepts, control statements, Rust ownership, references and borrowing, slices, Rust structs- update syntax and method syntax, functions and strings.	06
Unit-III	<b>Building blocks of Rust:</b> Rust Collection, enums in Rust, <b>Modules-</b> File System, visibility control, use keyword, introduction to standard library, using command-line and time, managing controlling system, using object-oriented concepts in Rust	06
Unit-IV	<b>Advance topics in Rust:</b> Handling errors, Vectors in Rust, Working with files and directories, rust generics, traits and lifetime. Using smart pointers in Rust. Implementing terminal I/O, processes and signals, Memory management.	08



Unit-V **Concurrency control in Rust:** Message passing through threads, error handling in threads, achieving concurrency with shared state, using threads with timers 07

Self-Study: The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents

- Suggested Readings/References:
1. Claus Matzinger, Rahul Sharma, and Vesa Kaihlavirta: The Complete Rust Programming Reference Guide
  2. Jason Orendorff and Jim Blandy: Programming Rust: Fast, Safe Systems Development
  3. Carol Nichols and Steve Klabnik: The Rust Programming Language (Covers Rust 2018)
  4. <https://doc.rust-lang.org/stable/book/>
  5. Antoni Boucher and Guillaume Gomez: Rust Programming By Example

Suggested List of Experiments:	Sr. No.	Practical Title	Hours
	1	Installation of RestUp environment. Use of cargo in RUST projects.	02
	2	Explore the basics of Rust concepts such as primitive datatypes, flow control, functions, and variable bindings.	02
	3	Create basic games like puzzles using RUST fundamentals.	02
	4	Perform basic operations on graph data structure using RUST.	04
	5	Apply fundamentals of RUST to generate random numbers and use them in cryptography.	04
	6	Perform linear data structure operations on vectors like searching, sorting, traversal etc.	04
	7	Explore in-built standard library and write a program based in hash-map	02
	8	Create GUI based games in RUST.	04
	9	Use the concept of file handling to read and store the data with appropriate I/O functions.	02
	10	Write a program to pass message between two threads. Consider and handle all the errors that may occur during this process.	04

Suggested Case List: -NA-