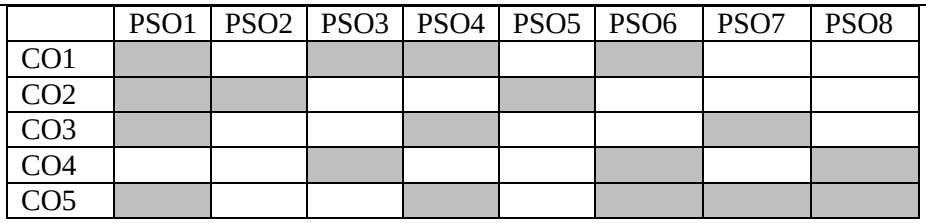
Course Code: 204
Course Title: Programming Skills

Course Code	204
Course Title	Programming Skills
Credits	4
Course Category	Major Course
Level of Course	200-299 (Intermediate Level)
Teaching per Week	4 Hours (2 Hours Theory + 4 Hours Practical)
Minimum weeks per Semester	15 (Including class work, examination, preparation etc.)
Review / Revision	2022-2023
Implementation Year:	A.Y. 2023-2024
Purpose of Course	To understand concepts of programming using Compiler based programming language C and Interpreter based programming Language Python. To compare the code structures of Compiler based programming language 'C' and interpreter based programming language 'Python'. [Python codes can be executed using any open source IDE. This is not IDE specific course.]
Course Objective	i) Advance programming skills using compiler based programming language C. ii) Introduction of Interpreter based Programming language Python. iii) Enhancing basic programming skills using Interpreter based and Compiler based programming languages
Pre-requisite	Fundamental knowledge of computer programming using 'C' language. Knowledge of Python IDE installation is recommended.
Course Outcomes	CO1: Students will be able to learn advanced programming concept of compiler based programming language. CO2: Students will be proficient working on conditional statements, iterative Statements and fundamentals of programming concepts using C and Python. CO3: Students will be able to represent compound data using lists, tuples and dictionaries in Python programs. CO4: Students will be able to develop real world application. CO5: Students will learn important libraries like Numpy, Pandas which are useful in Data analysis, Machine Learning.
Mapping between Course Outcomes(CO) with Program Specific Outcomes(PSO)	
Course Outcome	- On completion of the course, the Students will be conceptually clear about the two dimensional arrays, structures and unions using 'C' programming language. - Concept of conditional statements, iterative Statements and fundamentals of programming concepts using Python.
Course Content	**UNIT-1: Arrays, Structure & Union and User defined function in C programming Language** 1.1 Concepts of Two-Dimensional Numeric Array 1.1.1 Declaring Two-Dimensional numeric array 1.1.2 Two-Dimensional numeric Array operations (Addition, Subtraction, Multiplication, Transpose)

	1.1.3 Element Address in array(Row major and Column major) 1.1.4 Two-Dimensional Character Array: 1.1.4.1 Declaring& Initializing Two-Dimensional character array 1.1.4.2 Two-Dimensional character Array operations (Searching elements, copying, merging, finding length of given string) 1.2 Concepts of structure and Union 1.2.1 Defining, declaring and Initializing structure and Union 1.2.2 Typedef and accessing structure member 1.2.3 Difference between structure and union 1.3 User defined functions 1.3.1 Function return type, parameter list, local function variables 1.3.2 Passing arguments to function 1.3.3 Calling function from main() function or from other function. 1.3.4 Function with No arguments and no return value, No arguments and are turn value, with arguments and no return value, with arguments and are turn value. 1.3.5 Recursive Function **UNIT-2: Python Fundamentals** 2.1 Concepts of Interpreter based programming language 2.1.1 Structure of Python Programming language. 2.1.2 Python code Indention and execution 2.2 Python Variables 2.2.1 Naming of variables and Dynamic declaration of variables 2.2.2 Comments in Python 2.2.3 Assigning values to multiple variables 2.2.4 Global variables 2.3 Python Data types 2.3.1 Text(str), Numeric Type(int, float, complex),Boolean(bool) 2.3.2 Setting Data types 2.3.3 Type conversion(int, float, complex),casting(int, float, str) 2.4 User defined function. 2.4.1 Defining function, Function with Parameters 2.4.2 Parameter with default value, Function with return value **UNIT-3: Python Strings and Operators** 3.1 Python Strings 3.1.1 Multiline string, String as character array, triple quotes 3.1.2 Slicing string, negative indexing, string length, concatenation 3.1.3 String Methods: (centre, count, join, len, max, min, replace, lower, upper, replace, split) 3.2 Operators 3.2.1 Arithmetic Operators(+,-,*,/,%,**,//) 3.2.2 Assignment Operators(=,+=,-=,/=,*=,//=) 3.2.3 Comparison Operators (==,!=,>,<,>=,<=) 3.2.4 Logical Operators(and, or, not) 3.2.5 Identity and member operators(is, is not, in, not in) **UNIT-4: Python conditional and iterative statements** 4.1 If statement, if..elif statement, if..elif…else statements, nested if

	4.2 Iterative statements 4.2.1 While loop, nested while loop, break, continue statements. 4.2.2 for loop, range, break, continue, pass and Else with for loop, nested for loop. 4.3 List: creating list, indexing, accessing list members, range in list, List methods (append, clear, copy, count, index, insert, pop, remove, reverse, sort). **UNIT-5: Python Collections and Library** 5.1 Python Collections 5.1.1 Tuples: Declaring tuple, indexing tuple, changing tuple values, adding and removing data from tuple, Use of tuple() method to create tuple, count() and index() methods. 5.1.2 Sets: declaring set, access set data, set methods (add, clear, copy, discard, pop, remove, union, update). 5.1.3 Dictionary 5.1.3.1 Creating Dictionary, Adding, Accessing and Removing element 5.1.3.2 Dictionary methods: get(),pop(), popitem(),clear(),copy() 5.2 Introduction to Numpy and Pandas 5.2.1 Overview of numpy 5.2.1.1 Numpy methods (Mean, Median, Mode, Standard Deviation and Variance) 5.2.1.2 Implementation of Numpy methods on numeric data set created using list. 5.2.2 Pandas Dataframe 5.2.2.1 Creating dataframe using list 5.2.2.2 Creating dataframe using dict of equal length list 5.2.2.3 Reading data using csv file(read_csv()) 5.2.2.4 Retrieving rows and columns from data frame using index 5.2.2.5 Retrieving rows and columns using loc and iloc functions.
Reference Books	1.Programming in C, Balaguruswami - TMH 2. C Programming Language, Kernigham & Ritchie - TMH 3. The spirit of C, Cooper H & Mullish H - Jaico Pub. 4. Programming in C, Stephan Kochan - CBS 5. Mastering Turbo C, Kelly & Bootle - BPB 6. C Language Programming, Byron Gottfried –TMH 7. Learning Python -Mark Lutz : O'Reilly Media 8. Core Python Programming – by Wesley J Chun ISBN-13: 978- 0132269933 9. Python for Everybody: Exploring Data in Python 3, by Charles Severance (Author), Aimee Andrion (Illustrator), Elliott Hauser (Editor), Sue Blumenberg (Editor) 10. An Introduction to Python - by van Rossum Guido ISBN: 9780954161767, 0954161769 11. Core Pyhton Application Programming – by Wesley J Chun Prentice Hall
Teaching Methodology	Class Work, Discussion, Lab work, Self-Study, Seminars and/or Assignments
Evaluation Method	50% Internal assessment. 50% External assessment.