CSC2CJ101/CSC2MN100- Fundamentals Of Programming (C Language)

Programme	B. Sc. Computer Science				
Course Code	**CSC2CJ101/CSC2MN100**				
Course Title	**Fundamentals Of Programming (C Language)**				
Type Of Course	Major/Minor				
Semester	II				
Academic Level	100 – 199				
Course Details	Credit	Lecture per week	Tutorial per week	Practical per week	Total Hours
	4	3	-	2	75
Pre-requisites	1. Fundamentals of Algorithms and Flowcharts 2. CSC1CJ101 – Fundamentals of Computers and Computational Thinking				
Course Summary	The objectives of this course are to make the student understand programming language, programming, concepts of Loops, reading a set of Data, stepwise refinement, Functions, Control structure, Arrays, Structures, Unions, and Pointers. After completion of this course the student is expected to analyze the real life problem and write a program in 'C' language to solve the problem. The main emphasis of the course will be on problem solving aspect i.e. developing proper algorithms.				

Course Outcomes (CO):

CO	CO Statement	Cognitive Level*	Knowledge Category#	Evaluation Tools used
CO1	Remember the program structure of C with its syntax and semantics	U	C	Instructor-created exams / Quiz
CO2	Use the various constructs of a programming language viz. conditional, iteration and recursion.	Ap	P	Practical Assignment / Observation of Practical Skills
CO3	Use simple data structures like arrays and use the concepts of modular programming in solving problems.	Ap	C	Practical Assignment / Observation of Practical Skills
CO4	Demonstrate an understanding of storage classes, design and implement structured data types using structures and unions,and apply the concept of pointers for efficient memory management, dynamic data allocation in C Programming	Ap	P	Practical Assignment / Observation of Practical Skills
CO5	Develop efficient programs and implement algorithms in C programs for solving problem solving	Ap	P	Practical Assignment / Observation of Practical Skills Viva Voce

* - Remember (R), Understand (U), Apply (Ap), Analyse (An), Evaluate (E), Create (C)
# - Factual Knowledge(F) Conceptual Knowledge (C) Procedural Knowledge (P) Metacognitive Knowledge (M)

Detailed Syllabus:

Module	Unit	Content	Hrs
I	**Introduction to C Language**		**10**
	1	History of C, Importance of C, and sample programs	2
	2	Character set, Tokens, Constants, Variables, and Data types	2
	3	Operators - Arithmetic, Relational, logical, assignment, increment, decrement, conditional, bitwise and special operators. Arithmetic expressions, operator precedence, type conversions, mathematical functions	3
	4	Managing Input and Output Operators: Reading and writing a character, formatted input, formatted output.	3
II	**Decision Making Branching and Looping**		**10**
	5	Decision making with If - simple If, If else, nested If else, else If ladder	3
	6	Switch statement, conditional operator, Goto statement	2
	7	Loops: while, do while, for statements and nested loops	3
	8	Jumps in loops – break, continue	2
III	**Arrays and Functions**		**15**
	9	One dimensional array – declaration, initialization and accessing	2
	10	Two dimensional array – declaration, initialization and accessing	2
	11	Multi dimesnsional array, dynamic array	1
	12	Strings – Reading, Writing. Arithmetic operations on characters, Comparisons and string handling functions	2
	13	Functions – Need, Elements of user defined functions and definition	2
	14	Return values and their types, function call and declaration, call by value and call by reference	2
	15	Categories of functions, Nesting of functions	1
	16	Recursion and command line arguments	1
	17	Passing arrays to functions and passing strings to functions	2
IV	**Storage Classes, Structure and Union, Pointers**		**10**
	18	Storage classes – The scope, visibility and lifetime of variables. Auto, Extern, Static and Register storage classes. Storage classes in a single source file and multiple source files	2
	19	Structure and Union - Defining, giving values to members, initialization and comparison of structure variables, arrays of structure, arrays within structures, structures within structures, structures and functions, unions	2
	20	Pointers definition, declaring and initializing pointers, accessing a variable through address and through pointer, pointer expressions, pointer increments and scale factor	2
	21	Pointers and arrays, pointers and functions, pointers and structure	2
	22	Dynamic memory allocation and memory management functions	2
V	**Hands-on Problem Solving Using C** **Practical Applications, Case Study and Course Project**		**30**

	1	Implement the following: **1. Variables, Data types, Constants and Operators:** 1. Evaluation of expression ex: ((x+y) ^2 * (x+z))/w 2. Temperature conversion problem (Fahrenheit to Celsius) 3. Program to convert days to months and days (Ex: 364 days = 12 months and 4 days) 4. Salesman salary (Given: Basic Salary, Bonus for every item sold, commission on the total monthly sales) **2. Decision making (Branch / Loop) Statements:** 5. Solution of quadratic equation 6. Maximum of three numbers 7. Calculate Square root of five numbers (using goto statement) 8. Pay-Bill Calculation for different levels of employee (Switch statement) 9. Fibonacci series 10. Armstrong numbers 11. Pascal 's Triangle **3. Arrays, Functions and Strings:** 12. Prime numbers in an array 13. Sorting data (Ascending and Descending) 14. Matrix Addition and Subtraction 15. Matrix Multiplication 16. Transpose of a matrix 17Function with no arguments and no return value 18. Functions with argument and return value 19. Functions with argument and multiple return values 20. Function that convert lower case letters to upper case 21. Factorial using recursion. 22. Perform String Operations using Switch Case	30
		23. Largest among a set of numbers using command line argument **4. Structures and Union:** 24. Structure that describes a Hotel (name, address, grade, avg room rent, number of rooms) Perform some operations (list of hotels of a given grade etc.) 25. Using Pointers in Structures. 26. Cricket team details using Union. **5. Pointers:** 27. Evaluation of Pointer expressions 28. Function to exchange two pointer values 29. Reverse a string using pointers 30. Insertion, deletion, and searching in an array	

Mapping of COs with PSOs and POs:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO 1	3	1	2	3	1	1	3	0	3	1	3	0	0
CO 2	3	1	3	3	1	1	3	0	3	2	3	1	1
CO 3	3	1	3	3	2	2	3	0	3	3	3	1	1
CO 4	3	2	3	3	3	3	3	1	3	3	3	1	2
CO 5	3	2	3	3	3	3	3	1	3	3	3	1	2

Correlation Levels:

Level	Correlation
-	Nil
1	Slightly / Low
2	Moderate / Medium
3	Substantial / High

Mapping of COs to Assessment Rubrics:

	Internal Exam	Assignment	Project Evaluation	End Semester Examinations
CO 1	✓	✓		✓
CO 2	✓	✓		✓
CO 3	✓	✓		✓
CO 4	✓	✓		✓
CO 5	✓	✓	✓	✓

CSC2CJ102- Python Programming

Programme	B. Sc. Computer Science				
Course Code	CSC2CJ102				
Course Title	Python Programming				
Type of Course	Major				
Semester	IV				
Academic Level	200 - 299				
Course Details	Credit	Lecture per week	Tutorial per week	Practical per week	Total Hours
	4	3	-	2	75
Pre-requisites	1. CSC2CJ101 – Fundamentals of Programming				
Course Summary	This course explores the versatility of Python language in programming and teaches the application of various data structures using Python. The course also gives an introduction to scientific computing using popular Python packages.				

Course Outcomes (CO):

CO	CO Statement	Cognitive Level*	Knowledge Category#	Evaluation Tools used
CO1	Understand the basic concepts and problem solving skills using the basic constructs in python programming	Understand	C	Instructor-created exams / Quiz
CO2	Apply Modular Programming Using Functions in Python	Apply	P	Coding Assignments/ Code reading and review
CO3	Apply the various data structures and operations using python.	Apply	P	Coding Assignments/ Code reading and review
CO4	Apply various packages and visulization using python programming	Apply	C	Instructor-created exams / Case studies
CO5	Analyse using hands-on by practical programs, case studies and projects on different libraries of python	Analyse	P	Coding Assignments/ Case studies

* - Remember (R), Understand (U), Apply (Ap), Analyse (An), Evaluate (E), Create (C)
# - Factual Knowledge(F) Conceptual Knowledge (C) Procedural Knowledge (P) Metacognitive Knowledge (M)

Detailed Syllabus:

Module	Unit	Content	Hrs
I		Fundamentals of Python	12
	1	Features of Python, Identifiers, Keywords, Variables, Operators, Operands, Expressions and Data types	3
	2		
	3	Precedence and Associativity, Indentation, Comments	1
	4	Input, Output and Import functions, Mathematical functions, range function, Type Conversions	1
	5		3
	6	Decision-making Structures	3
		Looping Structures	1
		Control Statements	1
II		Functions & Modules	8
	7	Function Definition, Function Calling, Flow of Execution, Parameters and Arguments	2
	8		
	9	Types of Function Arguments – Required, Keyword, Positional and Variable length arguments	2
	10	Scope and lifetime of variables	1
	11	Types of Functions – Recursive, Anonymous, Functions with more than one return value, Void Functions	2
		Built in modules, User defined modules and packages	1
III		Data Structures in Python	15
	13	Strings - Indexing, Traversal, Slicing, Joining, and Splitting of Strings, Formatting Strings, Operation and Methods of Strings	5
	14		4
		Lists- Indexing and Traversal, Slicing, Joining, and Splitting of Lists, Operations and Methods of Lists	2
	15	Tuples – Indexing and Traversal, Operations and Methods of Tuples	3
	16	Dictionaries – Accessing and Modifying key-value pairs in Dictionary, Operations and Methods	1
			10
	17	Sets - Creation and Operations of Sets	2
IV		Introduction to Scientific Computing in Python	
	18	Introduction to NumPy Arrays – Advantage of NumPy Arrays, Creation of NumPy Arrays	3
	19		
	20	Computation on NumPy Arrays - Universal Functions, Broadcasting, Fancy Indexing	3
		Introduction to Pandas - Pandas Series and Pandas Data Frames. Series - Construction from arrays, explicit indices, and dictionaries. Data Frames- Construction from arrays and dictionaries.	
	21	Introduction to Matplotlib Basic plotting - Line plots, Scatter plots, Bar plots ,Histograms and Pie charts.	2
V		Hands-on Data Structures: Practical Applications, Case Study and Course Project	30

	1	Basics of Python 1. Demonstrate basic data types in python using interactive Interpreter. 2. Write a Python script that reads two integers and perform all arithmetic operations on these two numbers. 3. Write a program to compute distance between two points. 4. Write a program to calculate the area of a circle. Control Structures 5. Write a program to check whether a number is odd or even. 6. Write a program that reads a positive integer, n, from the user and then displays the sum of the first n natural numbers.	20
		7. Write a Python program to check whether a given year is a leap year or not. 8. Develop a program that reads a four-digit integer from the user and displays the sum of the digits in the number. For example, if the user enters 2151 then your program should display 2+1+5+1=9. Function 9. Write a program to find the largest of three numbers using functions. The program should pass three numbers as arguments and should return the result. 10. Write a function to check whether a given number is prime or not. 11. Write a recursive function to find the factorial of a number. Python Data Structures: Strings, Sets, Lists , Tuples and Dictionaries 12. Create a program that checks whether a given string is a palindrome or not. 13. Write a program to check whether an item exists in a tuple. 14. Write a program to create intersection, union, set difference, and symmetric difference of sets. 15. Write a program to create a telephone directory using a dictionary and display its contents. Also check for a specific phone number in the dictionary. NumPy, Pandas and Matplotlib 16. Write a program to implement matrix multiplication using NumPy. 17. Create a pandas series from a dictionary of values, and an ndarray. 18. Write a program to draw a line plot for the given heights and weights of a group of people. height=[145,155,165,175,185,195] weight=[43, 56, 60,69, 78,95]	
	2	Case Study	3
	3	Capstone (/Course) Project: Build a practical application using any one package and demonstrate using visualization tools.	7

Mapping of COs with PSOs and POs :

Mapping of COs to Assessment Rubrics :

	Internal Exam	Assignment	Project Evaluation	End Semester Examinations
CO 1				
CO 2				
CO 3				
CO 4				
CO 5				
CO 6				

Reference Books:

1. Jose, Jeeva. Taming Python By Programming. Khanna Book Publishing, 2017. Print.
2. S, Gowrishankar, and A, Veena. Introduction to Python Programming. Chapman & Hall/CRC Press, 2018.
3. Downey, Allen. Think Python. Green Tea Press, 2nd ed. 2009
4. VanderPlas, Jake. Python Data Science Handbook: Essential Tools for Working with Data. United States, O'Reilly Media, 2016.
5. Stephenson, Ben. The Python Workbook. SPRINGER INTERNATIONAL PU, 2016.

CO 6	✓	✓	✓	✓

Programme	B. Sc. Computer Science				
Course Code	CSC2MN101				
Course Title	Foundations of C Programming				
Type of Course	**Minor**				
Semester	II				
Academic Level	100-199				
Course Details	Credit	Lecture per week	Tutorial per week	Practical per week	Total Hours
	4	3	-	2	75
Pre-requisites	1. Basic Computer Literacy 2. Basic Problem-Solving Skills				
Course Summary	This course teaches the basics of programming using the C language. C is a powerful and widely used programming language known for its efficiency and flexibility. Through this course, students will learn how to write, understand, and debug C code to solve various problems and build simple applications.				

Course Outcomes (CO):

CO	**CO Statement**	**Cognitive Level***	**Knowledge Category#**	**Evaluation Tools used**
CO1	Demonstrate a solid understanding of fundamental programming concepts	An	P	Instructor-created lab exams / Quiz
CO2	Develop effective problem-solving skills by applying algorithmic thinking and logical reasoning.	An	P	Problem-solving assessments
CO3	Gain proficiency in writing, compiling, debugging, and executing C programs to implement algorithms, solve	Ap	P	Modeling Assignments

	problems, and create applications.			
CO4	Learn techniques to write efficient and optimized C code, including memory management, algorithm design, and performance tuning, to produce high-quality and scalable software solutions.	Ap	P	Modeling Assignments/ / Case studies
CO5	Understand and apply software development practices such as modular programming, code documentation, and debugging techniques to write maintainable and robust C programs. Develop critical thinking skills by analyzing and evaluating C code, identifying errors and inefficiencies, and proposing solutions to improve code quality and performance.	Ap	P	Modeling Assignments/ / Case studies Hands-on exercises

* - Remember (R), Understand (U), Apply (Ap), Analyse (An), Evaluate (E), Create (C)

- Factual Knowledge(F) Conceptual Knowledge (C) Procedural Knowledge (P) Metacognitive Knowledge (M)

Detailed Syllabus:

Module	Unit	Content	Hrs	Marks
I	**Problem-solving and logical Thinking**		**10**	**15**
	1	Overview of computational thinking concepts. Definition of algorithm and its characteristics.Understanding the importance of algorithms in problem-solving	2	
	2	Algorithm Development: Steps involved in designing algorithms	2	
	3	Pseudocode as an intermediate step in algorithm development.	1	

	4	Understanding flowchart symbols and their meanings.Learning to represent algorithms using flowcharts.	2	
	5	Raptor as a precursor to text-based programming languages	2	
	6	Drawing simple flowcharts	1	
II	**Introduction to C**		**10**	**20**
	7	Structure of C program	2	
	8	C Character Set, Keywords, Identifiers	1	
	9	Data Types, Variables, Declarations, Symbolic Constants	2	
	10	Operators:Arithmetic, Logical, Relational & Equality, and Unary, Operator Precedence and Associativity	2	
	11	Library Functions, Comments	1	
	12	I/O functions- Formatted scanf() & printf().	2	
III	**Control Statements, Arrays & Strings**		**14**	**20**
	13	Selection Statements:if, if-else, switch	3	
	14	iteration: while, do while, for	4	
	15	Arrays: One dimensional and Two Dimensional(introduction only)	3	
	16	Strings: Basic string handling functions	2	
	17	Structure:Definition, Processing-period Operator, Union(Concepts only)	2	
1V	**User-defined Functions**		**11**	**15**
	18	Definition of function, Advantages, Understanding function prototypes, and declarations	3	
	19	Introduction to function definitions and function calls	3	
	20	Exploring function parameters: actual and formal parameters	2	
	21	Recursion	2	
	22	Pointers-declarations(Basic concept only)	1	
V	**Hands-on C:** **Practical Applications, Case Study and Course Project**		**30**	

	1	Write a C program using Variables and Data Types Write a C program using Arithmetic Operations Write a C program using Loops Write a C program using Arrays Write a C program using Functions Write a C program using Strings	20	
	2	Case study: 1. Library Management System: Develop a program to manage a library's collection of books. Implement functions for adding, removing, and searching for books. 2. Ticket Booking System: Design a program to manage ticket bookings for a cinema or theater.	5	
	3	Capstone/Course Project: Design a real-time project in C	5	

Reference:

1. Balagurusamy, E. Programming in ANSI C. Tata McGraw-Hill Education, 2019.
2. King, K. N. C Programming: A Modern Approach. 2nd ed., W. W. Norton & Company, 2008.
3. Kernighan, Brian W., and Dennis M. Ritchie. The C Programming Language. 2nd ed., Prentice Hall, 1988.
4. Prata, Stephen. C Primer Plus. 6th ed., Addison-Wesley, 2013.
5. Perry, Greg. Absolute Beginner's Guide to C. 3rd ed., Que Publishing, 2014.
6. Oualline, Steve. Practical C Programming. 3rd ed., O'Reilly Media, 1997.
7. Hanly, Jeri R., and Elliot B. Koffman. Problem-Solving and Program Design in C. 8th ed., Pearson, 2016.
8. Gottfried, Byron S. Programming with C. 2nd ed., McGraw-Hill, 1996.
9. Holmes, Dan. C in a Nutshell. 2nd ed., O'Reilly Media, 2015.

Mapping of COs with PSOs and POs :

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	-	-	-	-	-	-	3	0	3	2	3	0	1
CO2	-	-	-	-	-	-	3	0	3	2	3	0	1
CO3	-	-	-	-	-	-	3	0	3	2	3	0	1
CO4	-	-	-	-	-	-	3	0	3	2	2	0	1
CO5	-	-	-	-	-	-	3	0	3	3	2	0	1

Correlation Levels:

Level	**Correlation**
-	Nil
1	Slightly / Low
2	Moderate / Medium
3	Substantial / High

Assessment Rubrics:

- Quiz / Assignment/ Quiz/ Discussion / Seminar
- Midterm Exam
- Programming Assignments (20%)
- Final Exam (70%)

Mapping of COs to Assessment Rubrics :

	Internal Exam	Assignment	Project Evaluation	End Semester Examinations
CO 1		✓		✓
CO 2	✓	✓		✓
CO 3		✓		✓
CO 4	✓			✓
CO 5	✓		✓	✓

Programme	B. Sc. Computer Science				
Course Code	CSC2MN102				
Course Title	Introduction to Data Science				
Type of Course	**Minor**				
Semester	II				
Academic Level	100-199				
Course Details	Credit	Lecture per week	Tutorial per week	Practical per week	Total Hours
	4	3	-	2	75
Pre-requisites	1. Python Programming 2. Linear Algebra				
Course Summary	This course provides a comprehensive overview of data science, covering the various types of data and their applications. The students will acquire a deep understanding of exploratory data analysis along with hands-on implementation skills. . The curriculum introduces both supervised and unsupervised and techniques of Machine learning. Additionally, the data pre-processing techniques are introduced Overall, the course provides a comprehensive understanding of the fundamental data science principles, guiding students through the data science process and illustrating practical applications.				

Course Outcomes (CO):

CO	CO Statement	Cognitive Level*	Knowledge Category#	Evaluation Tools used
CO1	Understand the types of data and the applications of data science	U	C	Instructor-created exams / Quiz
CO2	Implement various data analysis and visualisation techniques.	Ap	P	Problem-solving assessments

CO3	Express and Gain insights on advanced data preprocessing techniques	U	P	Modelling Assignments
CO4	Discuss various machine learning techniques and algorithms	U	P	Modelling Assignments/ / Case studies
CO5	Create analysis and prediction models using Machine learning methodologies	C	M	Modelling Assignments/ / Case studies Instructor-created exams / Quiz

* - Remember (R), Understand (U), Apply (Ap), Analyse (An), Evaluate (E), Create (C)

- Factual Knowledge(F) Conceptual Knowledge (C) Procedural Knowledge (P) Metacognitive Knowledge (M)

Detailed Syllabus:

Module	Unit	Content	Hrs	Marks (70)
I	**Introduction to Data Science**		**10**	**10**
	1	Introduction to Data: Types of Data – Structured Data, Semi-Structured Data, Unstructured Data and Data Streams, Statistical Data Types - Quantitative Data (Ratio and Interval Scale) and Qualitative Data (Nominal and ordinal)	2	
	2	Basic Methods of Data Analysis- Descriptive Data Analysis, Diagnostic Data Analysis or Exploratory Data Analysis, Inferential Data Analysis and Predictive Analysis.	1	
	3	Inferential Statistics: Statistical Inference, Population and Sample, Statistical Modeling, Probability Distributions – Normal, Uniform	3	
	4	Introduction to Data Science: Big Data and Data Science , Data Science Process	2	
	5	Applications of Data Science , Issues and challenges in Data Science	2	

II	**Exploratory Data Analysis**	**14**	**10**

	6	Exploratory Data Analysis (EDA) : Types of EDA - Univariate non-graphical, Multivariate non- graphical, Univariate graphical, and, Multivariate graphical. Methods of EDA – Descriptive Statistics and Data Visualization	5	
	7	Descriptive Statistics - Measures of Central Tendencies, Dispersion, Skewness and Kurtosis.	5	
	8	Data Visualization - Histograms , Box plots , Quantile-Quantile plots Scatter plots , Heat map, Bubble chart , Bar chart, Distribution plot , Pair plot , Line graph , Pie chart, Area chart	4	
III	**Data Preparation for Analysis**		**6**	**15**
	9	Data Cleaning: Handling Missing and Noisy Data, Removing outliers	2	
	10	Data Integration	1	
	11	Data Transformation: Standardization, Normalization	2	
	12	Data Reduction: Dimensionality Reduction - Principal Component Analysis	1	
1V	**Introduction to Machine Learning**		**15**	**15**
	13	Machine Learning Algorithms : Supervised Learning– Classification, Regression, Unsupervised Learning – Clustering, Dimensionality Reduction , Reinforcement Learning	3	
	14	Test /Train Split, Model Training, Bias and Variance, Overfitting and Underfitting	3	
	15	Evaluation	2	
	16	Linear Regression	1	
	17	k-Nearest Neighbors (k-NN)	1	
	18	k-means Clustering	1	
	19	Naive Bayes	1	
	20	Application of Naive Bayes - Spam Filtering	1	
	21	Singular Value Decomposition	1	
	22	Applications of Supervised, Unsupervised and Reinforcement	1	

		Learning		
V	**Hands-on Data Structures:** **Practical Applications, Case Study and Course Project**		**30**	**20**
	1	Implementation of the concepts or the algorithms learned [Binary Classification, Linear Regression, k-NN, k-means clustering, Spam Filtering]	15	
	2	Case study: Perform exploratory data analysis on a real world dataset using Python. Using appropriate Python packages parse, clean and visualize the data .	5	
	3	Capstone/Course Project: Perform an end-to-end project of the data science process.		

Mapping of COs with PSOs and POs :

	PSO1	PSO2	PSO3	PSO4	PSO 5	PSO 6	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO 1							3	1	2	3	1	1	1
CO 2							3	1	2	3	3	2	3
CO 3							3	1	2	3	3	2	3
CO 4							3	1	2	3	3	2	3
CO 5							3	1	2	3	3	2	3

Correlation Levels:

Level	**Correlation**
-	Nil

1	Slightly / Low
2	Moderate / Medium
3	Substantial / High

Assessment Rubrics:

- Quiz / Assignment/ Quiz/ Discussion / Seminar
- Midterm Exam
- Programming Assignments (20%)
- Final Exam (70%)

Mapping of COs to Assessment Rubrics :

	Internal Exam	Assignment	Project Evaluation	End Semester Examinations
CO 1		✓		✓
CO 2	✓	✓		✓
CO 3		✓		✓
CO 4	✓			✓
CO 5	✓		✓	✓

References

1. O'Neil, Cathy, and Rachel Schutt. *Doing data science: Straight talk from the frontline*. " O'Reilly Media, Inc.", 2013.
2. Han, Jiawei, et al. Data Mining: Concepts and Techniques. Netherlands, Elsevier Science, 2011.
3. Shah, Chirag. A Hands-On Introduction to Data Science. United Kingdom, Cambridge University Press, 2020.
4. Chopra, Rohan, et al. Data Science with Python: Combine Python with Machine Learning Principles to Discover Hidden Patterns in Raw Data. United Kingdom, Packt Publishing, 2019.

Programme	B. Sc. Computer Science				
Course Code	CSC2MN103				
Course Title	**Fundamentals of SPSS and R programming**				
Type of Course	**Minor**				
Semester	II				
Academic Level	100-199				
Course Details	Credit	Lecture per week	Tutorial per week	Practical per week	Total Hours
	4	3	-	2	75
Pre-requisites	1. Basic computer knowledge 2. Spreadsheet essentials				
Course Summary	This course offers SPSS basics including data management, transformation, visualization and statistical analysis techniques. Also inAtroduces fundamentals of R environment, focusing on data manipulation and visualization.				

Course Outcomes (CO):

CO	**CO Statement**	**Cognitive Level***	**Knowledge Category#**	**Evaluation Tools used**
CO1	Summarize essential data input and manipulation activities within SPSS..	U	P	Instructor-created exams / Seminar Presentation/ Instructor-created exams/ Quiz
CO2	Apply descriptive statistics and conduct parametric and nonparametric tests in SPSS	Ap	P	Assignment / Instructor-created exams
CO3	Summarize overview of R Environment	U	P	Assignment / Instructor-created exams
CO4	Describe Parametric and Non Parametric testing of Statistical Hypothesis	U	P	Hands-on practical sessions

CO5	Create effective visualizations using SPSS and R.	C	P	

* - Remember (R), Understand (U), Apply (Ap), Analyse (An), Evaluate (E), Create (C)

- Factual Knowledge(F) Conceptual Knowledge (C) Procedural Knowledge (P) Metacognitive Knowledge (M)

Detailed Syllabus:

Module	Unit	Content	Hrs	Marks
I	**Introduction to SPSS**		**12**	**19**
	1	Features – Data View – Variable View – Output Viewer Window – Syntax Editor Window -	1	
	2	Open data file , Save , import from other data source ,data entry , labelling for dummy numbers	2	
	3	Recode in to same variable, Recode in to different variable, Transpose of data, Insert variables and cases	2	
	4	Merge variables and cases, Split, Select cases, Compute total scores	2	
	5	Table looks – Changing column - font style and sizes	2	
	6	Diagrammatic representation	2	
II	**Data Analysis Using SPSS**		**10**	**18**
	7	Estimation of mean, median and mode- Standard deviation and coefficient of variation.	2	
	8	Descriptive statistics, Parametric tests t-test (paired or unpaired), ANOVA (one-way- two way)	2	

	9	Pearson rank correlation, Linear regression	3	
	10	Non parametric tests: Mann Whitney U test, Wilcoxon signed rank test .	2	
	11	Kruskall Wallis test ,Chi- Square test	1	
III	**Overview of R Environment**		**11**	**18**
	12	R editor, Workspace	2	
	13	Data type – Importing and Exporting Data	2	
	14	Basic Computational Ideas – Merges in R. Matrix Determinant – Inverse – Transpose,Trace)	3	
	15	Eigen Values and Eigen Vectors	2	
	16	Construction of Bar, Pie, Histogram, Line Chart, Box Plot, Scatter Plot	2	
IV	**Parametric and Non Parametric testing of Statistical Hypothesis**		**12**	**15**
	17	One Sample t test, Two group t tests, Paired t test, one way ANOVA, two way ANOVA	3	
	18	Wilcoxon, Mann Witney, Kruskal Wallis Simple Correlation	3	
	19	Linear Regression, Multiple Linear Regression, Testing for overall significance of Model Coefficients – Testing for Individual Regression Coefficients.	2	
	20	Outliers Detection Control Charts, Variable Control Chart, x, R, S.	2	
	21	Attribute Control Chart - p, np, c, u. CUSUM Control Chart, EWMA Control Chart.	2	

	22	Process Capability Analysis, Process Capability Analysis		
V	**Hands-on Word Processor and Presentation Tool:** **Practical Applications, Case Study and Course Project**		**30**	
		SPSS 1. Descriptive Statistics 2. Paired –Samples T Test 3. One-Way ANOVA 4. Correlation & Linear Regression 5. Chi- Square Test **R PROGRAMMING** 6. Simple Correlation 7. Linear Regression 8. One- Way ANOVA 9. Paired T test 10. Plotting Bar Chart	20	
		Case study(Example): **SPSS and R** 1. **Case Study: Customer Satisfaction Analysis** Analyze factors influencing customer satisfaction using survey data. Employ SPSS for regression analysis to identify significant predictors such as product quality, pricing, and customer service. Use R programming to analyse data and make predictions.	10	

Reference Books:

1. Michael S. Louis – Beck (1995). Data analysis an introduction, Series: quantitative applications in the social sciences. Sage, Publications. London

2. Jeremy J. Foster (2001). Data analysis using SPSS for windows. New edition. Versions 8-10. Sage publications. London.

3. Sprankle , M., Problem Solving & Programming Concepts, Pearson India
4. Learning Statistics using R By Rndall E.Schumacker, Sage Publication
5. R for Everyone By Jared P.Lander, Pearson Education

Mapping of COs with PSOs and POs :

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO 1	-	-	-	-	-	-	1	1	3	3	1	1	1
CO 2	-	-	-	-	-	-	2	1	3	3	3	1	1
CO 3	-	-	-	-	-	-	1	1	3	3	3	1	1
CO 4	-	-	-	-	-	-	1	1	3	3	3	1	1
CO 5	-	-	-	-	-	-	1	2	3	3	3	1	2

Correlation Levels:

Level	**Correlation**
-	Nil
1	Slightly / Low
2	Moderate / Medium
3	Substantial / High

Assessment Rubrics:

- Quiz / Assignment/ Quiz/ Discussion / Seminar
- Midterm Exam
- Programming Assignments (20%)
- Final Exam (70%)

Mapping of COs to Assessment Rubrics :

	Internal Exam	Assignment	Project Evaluation	End Semester Examinations
CO 1	✓	✓		✓
CO 2	✓	✓		✓
CO 3	✓	✓		✓
CO 4	✓	✓	✓	✓
CO 5	✓	✓	✓	✓