AC/24.02.2024/RS1

SIES College of Arts, Science & Commerce (Autonomous)

RISE WITH EDUCATION

NAAC REACCREDITED - 'A' GRADE

SIES College of Arts, Science and Commerce (Autonomous)

Affiliated to University of Mumbai

Syllabus under NEP effective from June 2024

Programme: B.Sc.

Subject: Information Technology

Core Course

Class: SYBSc(IT)

Semester : III and IV

Choice Based Credit System (CBCS)
with effect from the academic year 2024-25

Semester III
Core Course

This Core course is offered to students of BSc(IT) in Semester III, who have chosen Information Technology as Major & Minor subject

Name of Programme: Bachelor of Science **Subject:** Information Technology						
Class	**Semester**	**Course Code**	**Course Name**	**No. of Lectures/ Practicals per week**	**Credits**	**Marks**
SYBSc(IT)	III	SIUITMJ211	Python Programming - I	3L	3	75
SYBSc(IT)	III	SIUITMJP211	Python Programming - I Practical	1P per batch	1	25
P (Practical) = 2 Hours per week						

Course Name: Python Programming - I **Credits: 3 Type: Theory**		
Expected Course Outcomes On completion of this course, students will be able to 1. Write programs using the conditional statements and loops in Python 2. Explain the importance of functions and apply various operations on strings in Python. 3. Classify the different data structures, lists, tuples, and dictionaries, in Python.		
Pre-requisites:	Basics of Programming	
Unit I	Conditional Statement and Loops	15 Lectures
	● Variables and Expressions: Values and Types, Variables, Variable Names and Keywords, Type conversion, Operators and Operands, Expressions, Interactive Mode and Script Mode, Order of Operations. ● Conditional Statements: if, if-else, nested if –else ● Looping: for, while, nested loops ● Control statements: Terminating loops, skipping specific conditions.	
Unit II	Functions and Strings	15 Lectures

	• Functions: Function Calls, Type Conversion Functions, Math Functions, Composition, Adding New Functions, Definitions and Uses, Flow of Execution, Parameters and Arguments, Variables and Parameters Are Local, Stack Diagrams, Fruitful Functions and Void Functions, Why Functions? Importing with from, Return Values, Incremental Development, Composition, Boolean Functions, More Recursion, Leap of Faith, Checking Types • Strings: A String Is a Sequence, Traversal with a for Loop, String Slices, Strings Are Immutable, Searching, Looping and Counting, String Methods, The in Operator, String Comparison, String Operations	
Unit III	Data Structures, Files and Exceptions	15 Lectures
	• Lists: Values and Accessing Elements, Lists are mutable, traversing a List, Deleting elements from List, Built-in List Operators, Concatenation, Repetition, In Operator, Built-in List functions and methods • Tuples and Dictionaries: Tuples, Accessing values in Tuples, Tuple Assignment, Tuples as return values, Variable-length argument tuples, Basic tuples operations, Concatenation, Repetition, in Operator, Iteration, Built-in Tuple Functions Creating a Dictionary, Accessing Values in a dictionary, Updating Dictionary, Deleting Elements from Dictionary, Properties of Dictionary keys, Operations in Dictionary, Built-In Dictionary Functions, Built-in Dictionary Methods	

Course Name: Python Programming – I Practical
Credits: 1 Type: Practical

Expected Course Outcomes

On completion of this course, students will be able to

1. Apply the use of if structure, loops and functions and execute them by writing programs.
2. Compare and Test the use of various data structures in Python.

Practical No.	Title
01	Write Python program to do the following: Use Function a. Accept the name and age of the user. Print a message addressing the user the year they will turn 100 years old. For eg. If the name and age is entered as Nimith and 20, the o/p will be "Nimith, you will turn 100 in the year 2102" b. Accept 2 numbers from the user and swap the numbers, with and without using multiple assignment statement. For eg. If a=10 and b=20 before swapping then the o/p will be a=20 and b=10 after swapping. c. Generate Fibonacci series with 10 terms, with and without using multiple assignment statement. Output will be 0 1 1 2 3 5 8 13 21 34 d. Accept a number and Reverse the number, also check if it is a Palindrome, use return statement. For eg. If the i/p is 12345 then output will be 54321. Eg. for Palindrome say for i/p 1221 o/p is "1221 is a Palindrome", for i/p 1234 o/p "1234 is not a Palindrome" e. Accept a number and check if the given number is an Armstrong number or not. For eg. 153 = 13 + 53 + 33 f. Accept a number and check if the given number is a Strong number or not. For eg. 145 = 1! + 4! + 5! Accept a number. Write a recursive function to print the factorial of the number. For eg. 3! = 6
02	Write Python program to do the following: Use String a. Accept a string. Remove the characters at the odd index and print the original string and the transformed string. For eg. If original string S = "SIESASCS" the transformed string will be "SEAC" b. Accept a list of words and return the longest word and the length of the longest word. For eg. given a list L = ["Truth", "Determination", "Perseverance", "Freedom", "Faithful", "Courage", "Hope"] the o/p will be "Longest word is "Determination" and length is 13". c. Accept a comma separated sequence of words as input and print the words in sorted form (alphanumerically). For eg. i/p may be given as Truth, Determination, Perseverance, Freedom, and the o/p will be Determination, Freedom, Perseverance, Truth d. Count occurrences of a substring in a string. For eg. Given a string "Fear leads to anger; anger leads to hatred; hatred leads to conflict; conflict leads to suffering.", if requested to find the occurrences of the substring "hatred" the o/p should be 2. e. Reverse words in a string. For eg. Given a string "Don't take rest after your first victory

	because if you fail in second, more lips are waiting to say that your first victory was just luck." The o/p will be luck just was victory first your that say to waiting are lips more second, in fail you if because victory first your after rest take Don't f. A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Write a program to check whether the given sentence is a pangram or not. g. Count repeated characters in a string. For eg. Given a string "I have stood on a mountain of no's for one yes" the o/p should be 10 o 8 n 5 a 3 e 3 s 3 t 2 f 2 g. Convert a given string into a list of words. For eg. Given a string "If there is no struggle, there is no progress." The o/p will be ['If', 'there', 'is', 'no', 'struggle,', 'there', 'is', 'no', 'progress.'] h. Count and display the count of vowels and the vowels in a given text. For eg. Given a string "When one door of happiness closes, another opens; but often we look so long at the closed door that we do not see the one which has been opened for us." The o/p should be count of vowels is 48 and the vowels are ['e', 'o', 'e', 'o', 'o', 'o', 'a', 'i', 'e', 'o', 'e', 'a', 'o', 'e', 'o', 'e', 'u', 'o', 'e', 'e', 'o', 'o', 'o', 'o', 'a', 'e', 'o', 'e', 'o', 'o', 'a', 'e', 'o', 'o', 'e', 'e', 'e', 'o', 'e', 'i', 'a', 'e', 'e', 'o', 'e', 'e', 'o', 'u'] Remove spaces from a given string. For eg. Given a string "Do one thing every day that scares you." The o/p will be Doonethingeverydaythatscaresyou.
03	Write the Python program to do the following: Use List a. Write a program that takes two lists and returns True if they have at least one common member and returns None if no common member is found. For eg. Given the 2 lists [1,2,3,4,5], [5,6,7,8,9] o/p will be True, given i/p [1,2,3,4,5], [6,7,8,9] the o/p will be None b. Write a program to print a specified list after removing the 1st, 2nd, 5th elements. For eg. Given the list ['Action', 'Believe', 'Commitment', 'Confidence', 'Dare', 'Focus'] o/p will be ['Action', 'Commitment', 'Dare'] Write a programs to print the Fibonacci series. c. Accept a list from the user. Write a program to get the largest and smallest number in the list. d. Write a program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings. For eg. Given the list ['a','charismatic', 'champion', 'beaming', '4554','dignified'] the o/p will be 3 e. Write a program to shuffle and print a specified list. For eg. Given the list ['Action', 'Believe', 'Commitment', 'Confidence', 'Dare', 'Focus'] o/p will be ['Commitment', 'Focus', 'Action', 'Believe', 'Dare', 'Confidence'] Write a program to select an item randomly from a list. Given the list ['Action', 'Believe', 'Commitment', 'Confidence', 'Dare', 'Focus'] o/p will be ['Commitment'] (Note :user has no control over the choice)

04	Write the program to do the following: Use Tuple a. Write a program to convert the given list to a tuple. For eg. Given the list [45, 23, 67, 12, 7, 99] o/p will be (45, 23, 67, 12, 7, 99). b. Write a program to calculate the product of all the numbers in a given tuple. For eg. Given the tuple (45, 23, 22, 2, -16, 19) the o/p will be -13844160 c. Print all pair combinations of given 2 tuples. For eg. Given tuple 1 : (1, 3) and tuple 2 : (4, 9) the o/p , the combined tuple will be : [(1, 4), (1, 9), (3, 4), (3, 9), (4, 1), (4, 3), (9, 1), (9, 3)] Test if tuple is distinct. For eg. Given tuple : (13, 54, 95, 76, 11, 54) o/p will be False, given tuple : (13, 54, 95, 76, 11, 584) o/p will be True
05	Write the program to do the following: Use Dictionary a. Write a Python script to sort (ascending and descending) a dictionary by key and by value. For eg. Given the dictionary {1:2, 3:4, 4:3, 2:0, 0:0} o/p will be Original Dictionary : {1: 2, 3: 4, 4: 3, 2: 0, 0: 0} Ascending order by Value : {2: 0, 0: 0, 1: 2, 4: 3, 3: 4} Descending order by Value : {3: 4, 4: 3, 1: 2, 2: 0, 0: 0} Ascending order by Key : {0: 0, 1: 2, 2: 0, 3: 4, 4: 3} Descending order by Key : {4: 3, 3: 4, 2: 0, 1: 2, 0: 0} b. Create grade calculator. For eg. Given the dictionaries # 1. Trusha's dictionary trusha = { "name":"Trusha Salian", "assignment" : [90, 90, 90, 90], "test" : [95, 95], "lab" : [90.10, 90.10]} # 2. Chrishanth's dictionary chrishanth = { "name":"Chrishant Lukshmanraj", "assignment" : [82, 76, 48, 50], "test" : [90, 90], "lab" : [66.30, 66.42] } o/p will be Trusha Salian

	---------- Average marks of Trusha Salian is : 93.52 Grade of Trusha Salian is : A Chrishant Lukshmanraj ---------- Average marks of Chrishant Lukshmanraj is : 82.672 Grade of Chrishant Lukshmanraj is : B Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys. The o/p will be {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}

References

1. Think Python by Allen Downey, Published by O'Reilly , 2nd Edition , 2015
2. An Introduction to Computer Science using Python 3, by Gries, Paul, Campbell, Jennifer, Montojo, Jason , SPD Publications, 1st Edition 2014
   https://www.python.org/

Scheme of Evaluation:

I) Continuous Internal Evaluation (25 Marks)	
Class Test	10 Marks
Assignment/ Project and Viva/ Presentation	15 Marks
II) Theory Examination (50 Marks)	
-Semester End Examination based on entire syllabus	50 Marks
III) Practical Examination (25 Marks)	
Certified Journal	5 Marks
Viva Voce	5 Marks
Practical exam	15 Marks

Semester III
Core Course

This Core course is offered to students of BSc(IT) in Semester III, who have chosen Information Technology as Major & Minor subject

Name of Programme: Bachelor of Science			**Subject:** Information Technology			
Class	**Semester**	**Course Code**	**Course Name**	**No. of Lectures/ Practicals per week**	**Credits**	**Marks**
SYBSc(IT)	III	SIUITMJ212	Database Management Systems - I	3L	3	75
SYBSc(IT)	III	SIUITMJP212	Database Management Systems - I Practical	1P per batch	1	25
P (Practical) = 2 Hours per week						

Course Name: Database Management Systems - I

Credits: 3 Type: Theory

Expected Course Outcomes

On completion of this course, students will be able to

1. Examine and conceptualize data using the relational model and create Entity Relationship diagrams for data models.
2. Explain the ACID properties of transactions, different types scheduling in transactions, concurrency control in DBMS.
3. Use SQL to create, manage the database objects in the database and retrieve data from the database.

Unit I	**Introduction to Databases and Transactions** What is database system, purpose of database system, view of data, relational databases, database architecture, transaction management **Data Models** The importance of data models, Basic building blocks, Business rules **Database Design, ER Diagram** Database design and ER Model: overview, ER Model, Constraints, ER Diagrams, ERD Issues, weak entity sets.	15 Lectures
Unit II	**Relational database model:** Logical view of data, keys, integrity rules, Relational Database design: features of good relational database design, atomic domain and Normalization (1NF, 2NF, 3NF, BCNF) with cases. **Transaction management and Concurrency** Control Transaction management: ACID properties,	15 Lectures

	serializability and concurrency control, Lock based concurrency control (2PL, Deadlocks), Time stamping methods, optimistic methods, database recovery management.	
Unit III	**SQL:** Introduction, Relational Databases,SQL Basics, **Simple Queries** : SELECT statement, WHERE clause, Search conditions in a WHERE clause, Multitable Queries, **Summary Queries:** Column functions, Grouped Queries, Group Search Conditions **SQL : Subqueries:** Using subqueries, Subquery search conditions, subqueries and joins, Nested subqueries, Correlated subqueries, **Creating Database :**Create, Alter Drop statements ,Database Updates : Insert, Update ,Delete statements, Data Integrity, Views, SQL Security	15 Lectures

Course Name: Database Management Systems - I Practical
Credits: 1 Type : Practical

Expected Course Outcomes

On completion of this course, students will be able to

1. Write SQL queries to create and manage table structures and their data
2. Implement joins, aggregations, subqueries using SQL
3. Use SET operators, advanced group by queries and hierarchical queries in SQL

Practical No.	Title
1.	**SQL Statements – 1**
	a. Writing Basic SQL SELECT Statements
	b. Restricting Data – WHERE clause
	c. Sorting Data – ORDER BY clause
2.	**SQL Statements – 2**
	a. Displaying Data from Multiple Tables – Equi joins,Self joins,Outer joins
	b. Summary Functions and Group BY clause
	c. Subqueries
3.	**Creating and Managing Tables**
	a. Creating and Managing Tables : Use CREATE,ALTER DROP statements to create a set of tables that form a part of a database.(Handle Referential Cyles)
	b. Including Constraints : Primary key, foreign key, Not null, Check
4.	**Manipulating Data**
	Using INSERT, UPDATE, DELETE statement to insert rows,update rows and delete rows in the tables created in practical 3 (Handle referential cycles)
5.	**Creating and Managing other database objects**
	a. Creating Views: Create view , with check option
	b. Creating Sequences, Synonyms, Index
6.	**Using SET operators, GROUP BY clause (advanced features),hierarchical retrieval**
	a. Using SET Operators
	b. Enhancements to the GROUP BY Clause like Roll Up, Cube, Grouping Sets.
	c. Hierarchical retrieval queries

References

1. Database System and Concepts by A Silberschatz, H Korth, S Sudarshan, McGraw-Hill, 5th ed.
2. Database Systems by Rob Coronel, Cengage Learning, 12th ed.
3. SQL – The Complete Reference by Paul Weinberg,James Groff,Andrew Oppel, McGraw-Hill, 3rd ed.
4. Introduction to Database System by C.J.Date, Pearson,1st ed.

Scheme of Evaluation:

I) Continuous Internal Evaluation (25 Marks)	
Class Test	10 Marks
Assignment/ Project and Viva/ Presentation	15 Marks
II) Theory Examination (50 Marks)	
Semester End Examination based on entire syllabus	50 Marks
III) Practical Examination (25 Marks)	
Certified Journal	5 marks
Viva Voce	5 marks
Practical exam	15 marks

Semester III
Core Course

This Core course is offered to students of BSc(IT) in Semester III, who have chosen Information Technology as Major & Minor subject

Name of Programme: Bachelor of Science			**Subject:** Information Technology			
Class	**Semester**	**Course Code**	**Course Name**	**No. of Lectures/ Practicals per week**	**Credits**	**Marks**
SYBSc(IT)	III	SIUITMN211	Embedded Systems	3L	3	75
SYBSc(IT)	III	SIUITMNP211	Embedded Systems Practical	1P per batch	1	25
P (Practical) = 2 Hours per week						

Course Name: Embedded Systems – I **Credits: 3 Type: Theory**		
Expected Course Outcomes On completion of this course, students will be able to 1. Understand the organization and design of the 8051 microcontroller-based embedded systems. 2. Interface the input/output devices and the communication devices with the microcontroller. 3. Programme the microcontroller using embedded C language.		
Pre-requisites:	Basics of C language	
Unit I	**Introduction to Embedded Systems, Sensors and Actuators**	15 Lectures
	Introduction; • Embedded systems vs. General computing systems, Microprocessor vs. Microcontroller • Harvard vs. Von- Neumann processor/controller Architecture • RISC vs. CISC controller/Processor • Digital signal processor (DSP) **Sensors and Actuators:** • Light Emitting Diode • 7-Segment LED display, • Optocoupler, • Stepper motor, • Piezo buzzer, • Keyboard, etc.	

	Memory: • Program storage memory (ROM) • Read/Write Memory (RAM).	
Unit II	**Communication Interface and Designing Embedded systems with 8051 microcontrollers**	15 Lectures
	Communication Interface: • On-board communication interfaces • External communication interfaces **Designing Embedded systems with 8051 microcontrollers:** • 8-Bit Microcontroller • Architecture • Instruction Set and Programming • Programming Parallel Ports • Timers and Serial Port Registers, SFR Registers Interrupt Handling	
Unit III	**8051 programming in C**	15 Lectures
	• Data types and time delay • I/O Programming • Logic operations • Data conversion programs • Data serialization • Programming 8051 timers • Counter programming • Accessing code ROM space • Integrated Development Environment(IDE) • Assembling/Compiling and Execution	

Course Name: Embedded Systems Practical Credits: 1 Type : Practical	
Expected Course Outcomes On completion of this course, students will be able to 1. Programme a microcomputer to meet the requirements of the user. 2. Interface various peripherals and handle interrupts.	
Practical No.	Title
01	Introduction to Keil micro vision software
02	Program to store data in the Accumulator
03	Program to send values 00-ff to port 1
04	Program to toggle all the bits of P1 continuously
05	Program to create square/sine/triangular waves for 1ms delay.
06	Program to send data serially.
07	Program to simulate the binary counter.
08	Creating delay using four different timer modes.
09	Program to convert the hex to decimal and display digits on the ports.
10	Program to simulate stepper motor using Proteus.

References

1. Introduction to embedded systems by Shibu K V, Tata McgrawHill, 2012.
2. The 8051 Microcontroller and Embedded Systems by Muhammad Ali Mazidi, Pearson 2011

Scheme of Evaluation:

I) Continuous Internal Evaluation (25 Marks)	
Class Test	10 Marks
Assignment/ Project and Viva/ Presentation	15 Marks
II) Theory Examination (50 Marks)	
Semester End Examination based on entire syllabus	50 Marks
III) Practical Examination (25 Marks)	
Certified Journal	5 marks
Viva Voce	5 marks
Practical exam	15 marks

Semester IV
Core Course

This Core course is offered to students of BSc(IT) in Semester IV, who have chosen Information Technology as Major & Minor subject

Name of Programme: Bachelor of Science **Subject:** Information Technology						
Class	**Semester**	**Course Code**	**Course Name**	**No. of Lectures/ Practicals per week**	**Credits**	**Marks**
SYBSc(IT)	III	SIUITMJ221	Python Programming - II	3L	3	75
SYBSc(IT)	III	SIUITMJP221	Python Programming - II Practical	2P per batch	1	25
P (Practical) = 2 Hours per week						

Course Name: Python Programming - II **Credits: 3 Type: Theory**		
Expected Course Outcomes On completion of this course, students will be able to 1. Write programs using files and Exceptions in Python. 2. Find the various patterns in any text matter using Regular expressions in Python. 3. Examine the power of Object Oriented Programming using Python. 4. Illustrate GUI in Python and use MYSQL data in Python interface.		
Pre-requisites:	Basics of Programming	
Unit I	Files, Exceptions, Regular Expressions, Modules	15 Lectures
	● Files: Text Files, The File Object Attributes, Directories ● Exceptions: Built-in Exceptions, Handling Exceptions, Exception with Arguments, User-defined Exceptions ● Regular Expressions – Concept of regular expression, various types of regular expressions, using match function ● Modules: Importing module, Creating and exploring modules, Math module, Random module, Time module	
Unit II	Classes	15 Lectures
	● Classes and Objects: Overview of OOP (Object Oriented Programming), Class Definition, Creating Objects, Instances as Arguments, Instances as return values, Built-in Class Attributes, Inheritance, Method Overriding, Data Encapsulation, Data Hiding	

Unit III	GUI, MYSQL Database	15 Lectures
	● Creating the GUI Form and Adding Widgets: ➢ Widgets: Button, Canvas,Check button, Entry, Frame, Label, Listbox, Menu button, Menu, Message, Radio button, Scale, Scrollbar, text, Top level, Spinbox, Paned Window, LabelFrame, Messagebox. Handling Standard attributes and Properties of Widgets. ➢ Layout Management: Designing GUI applications with proper Layout Management features. ➢ Look and Feel Customization: Enhancing Look and Feel of GUI using different appearances of widgets. ● Data in Our MySQL Database via Our GUI: Connecting to a MySQL database from Python, Configuring the MySQL connection, Designing the Python GUI database, using the INSERT command, using the UPDATE command, using the DELETE command, Storing and retrieving data from MySQL database.	

Course Name: Python Programming – II Practical
Credits: 1 Type: Practical

Expected Course Outcomes

On completion of this course, students will be able to

1. Apply the use of files, modules, exceptions and execute them by writing programs.
2. Test the power of Object Oriented Programming.
3. Design a GUI and store and retrieve data from MYSQL database

Practical No.	Title
01	Write the program to do the following: Use File i. Write a program to read an entire text file ii. Write a program to append text to a file and display the text. Write a program to read last n lines of a file.
02	Write the program to do the following: Use Class a. Create a class named Parent with constructor to initialise firstname and lastname and a method to display the same. Derive a class Child from Parent, add a constructor here to initialise the age and Adhaar Card number and a method to display the same. Derive a class GrandChild from Child, add a constructor here to initialise the address and PAN Card number and a method to display the same. Write a driver code to implement Multilevel Inheritance. A polygon is a closed figure with 3 or more sides. Create a class called Polygon with data attributes to store the number of sides n and magnitude of each side as a list called sides. Add 2 methods, inputSides() method that takes in the magnitude of each side and a method dispSides() displays these side lengths. A square is a polygon with 4 sides. Create a class called square which inherits from Polygon. Define a method findAreaofsquare() to find and print the area of the square. Write a driver code to implement the above.
03	Write the program to do the following: a. Create a user defined module stropersodfib . Import the same in a Python (.py) code and use the functions defined in that user defined module and for the following: • Check for Prime number • Check for Perfect number • Find the sum of the digits of a given number • Print the Series 1, 20, 400, 8000, 160000 up to 10 terms. Use user defined modules. b. Accept the string "We God's creation! Worried about imperfection. Why lot of confusion? Which ends up in tension! Unable to pay attention? You are the one in control of the situation. Find a solution." as data. • Print the words ending in "tion" • Print the words starting with 'W or a' and count the same • Count the number of lines in the string • Count the number of words ending in "tion" and not ending in "tion" Use regular expressions.

	Write a program to implement exception handling. Demonstrate use of ZeroDivisionError and a user defined exception (a variable is uninitialized and is been used in the code , this should raise an exception say 'Variable not initialize").
04	Write the program for the following: Use Widget Demonstrate the use of the different Widgets say Label, Button, ComBox, CheckButton, RadioButton, Entry, Frame, Message in Python tkinter.
05	Design the database applications for the following: Create a database Student in SQL Server. Create a table Sdetails with columns Sid, Sname, Sage, Spnrnumber and Sdept . Sid is the Primary key of the table. Create a table Department with columns Deptid and Deptname. Deptid is the Primary key of the table. Sdept column in Sdetails table is the Foreign key whose values are derived from the Primary key column Deptid in Department table. Insert values into both the tables. Update values in the rows of the tables. Delete rows in the tables.

References

1. Python GUI Programming Cookbook by Burkhard A. Meier, Packt Publications, 2015
2. https://www.python.org/

Scheme of Evaluation:

I) Continuous Internal Evaluation (25 Marks)	
Class Test	10 Marks
Assignment/ Project and Viva/ Presentation	15 Marks
II) Theory Examination (50 Marks)	
Semester End Examination based on entire syllabus	50 Marks
III) Practical Examination (25 Marks)	
Certified Journal	5 Marks
Viva Voce	5 Marks
Practical exam	15 Marks

Semester IV
Core Course

This Core course is offered to students of BSc(IT) in Semester IV, who have chosen Information Technology as Major & Minor subject

Name of Programme: Bachelor of Science			**Subject:** Information Technology			
Class	**Semester**	**Course Code**	**Course Name**	**No. of Lectures/ Practicals per week**	**Credits**	**Marks**
SYBSc(IT)	IV	SIUITMJ222	Database Management Systems - II	3L	3	75
SYBSc(IT)	IV	SIUITMJP222	Database Management Systems - II Practical	1P per batch	1	25
P (Practical) = 2 Hours per week						

Course Name: Database Management Systems - I **Credits: 3 Type: Theory**		
Expected Course Outcomes On completion of this course, students will be able to 1. Describe variables and executable statements in PL/SQL. 2. Use loops, cursors, exceptions in PL/SQL. 3. Design and examine procedures, functions and triggers in PL/SQL.		
Unit I	**PL-SQL**: **Overview of PL/SQL:** About PL/SQL, Environment, Benefits, **Declaring Variables:** PL/SQL Block Structure, Block Types, Handling variables, Declaring variables, %TYPE, PUT_LINE procedure, **Writing Executable Statements:** PL/SQL Block syntax, Identifiers, Comments, Scope, Operators, **Interacting with Oracle Server :** SQL statements in PL/SQL, Naming Conventions, SQL Cursor and its attributes	15 Lectures
Unit II	**PL-SQL**: **Writing Control Structures :** Conditional IF statements, CASE expressions, LOOP statements, **Working with Composite Datatypes**: PL/SQL Records, %ROWTYPE, INDEX BY Table ,INDEX BY Table of Records, **Writing Explicit Cursors :** Steps to create and use explicit cursors, Explicit cursor attributes, Cursors and Records, Cursor For Loops, **Handling Exceptions:** Pre-defined, User-defined , RAISE_APPLICATION_ERROR,	15 Lectures
Unit III	**Procedures :** What is a procedure, syntax, modes, removing a procedure, **Functions:** What is a function, syntax, executing	15 Lectures

	functions, removing a function, Comparing procedures and functions, Packages: What are packages, components, creating a package, removing a package, Advantages, **Triggers:** What is a trigger, types, guidelines for designing triggers, Statement and Row triggers, Creation and use of DML triggers. Using conditional Predicates, Using OLD and NEW Qualifiers, Removing a Trigger.	

Course Name: Database Management Systems - II Practical
Credits: 1 Type : Practical

Expected Course Outcomes

On completion of this course, students will be able to

1. Use variables and executable statements in PL/SQL.
2. Use loops, cursors, exceptions in PL/SQL.
3. Design and examine procedures, functions and triggers in PL/SQL.

Practical No.	Title
1.	**PL/SQL Basics and Loops**
	a. Declaring Variables and Executing SQL statements in PL/SQL
	b. Writing Control Structures- Loops
2.	**Composite data types andcursors**
	a. Working with Composite Data Types – RECORD,INDEX BY
	b. Writing Explicit Cursors and CURSOR for loops
3.	**Procedures and Functions**
4.	**Exceptions and Packages**
5.	**Triggers**

References

1. Introduction to Oracle9i: PL/SQL:Volume I & II by Oracle.
2. Programming with PL/SQL for Beginners by H. Dand, R. Patil and T. Sambare, X-Team,1st ed.
3. Oracle PL/SQL Programming by Steven Feuerstein,O'Reilly,6th ed.

Scheme of Evaluation:

I) Continuous Internal Evaluation (25 Marks)	
Class Test	10 Marks
Assignment/ Project and Viva/ Presentation	15 Marks
II) Theory Examination (50 Marks)	
Semester End Examination based on entire syllabus	50 Marks
III) Practical Examination (25 Marks)	
Certified Journal	5 marks
Viva Voce	5 marks
Practical exam	15 marks

Semester IV
Core Course

This Core course is offered to students of BSc(IT) in Semester IV, who have chosen Information Technology as Major & Minor subject

Name of Programme: Bachelor of Science		**Subject:** Information Technology				
Class	**Semester**	**Course Code**	**Course Name**	**No. of Lectures/ Practicals per week**	**Credits**	**Marks**
SYBSc(IT)	IV	SIUITMN221	Computer Oriented Statistical Techniques	3L	3	75
SYBSc(IT)	IV	SIUITMNP221	Computer Oriented Statistical Techniques Practical	1P per batch	1	25
P (Practical) = 2 Hours per week						

Course Name: Computer Oriented Statistical Techniques

Credits: 3 Type: Theory

Expected Course Outcomes

On completion of this course, students will be able to

1. Explain the use of central tendency measures of data.
2. Apply the concepts of standard deviation, range, quartiles to understand the dispersion of the given data.
3. Compose Hypothesis and perform Hypothesis testing and take statistical decisions.

Pre-requisites:	None	
Unit I	• **The Mean, Median, Mode, and Other Measures of Central Tendency**: Index, or Subscript, Notation, Summation Notation, Averages, or Measures of Central Tendency ,The Arithmetic Mean , The Weighted Arithmetic Mean ,Properties of the Arithmetic Mean ,The Arithmetic Mean Computed from Grouped Data ,The Median ,The Mode, The Empirical Relation Between the Mean, Median, and Mode, The Geometric Mean G, The Harmonic Mean H ,The Relation Between the Arithmetic, Geometric, and Harmonic Means, The Root Mean Square, Quartiles, Deciles, and Percentiles, Software and Measures of Central Tendency. • **The Standard Deviation and Other Measures of Dispersion**: Dispersion, or Variation, The Range, The Mean Deviation, The Semi Interquartile Range, The 10–	15 Lectures

	90 Percentile Range, The Standard Deviation, The Variance, Short Methods for Computing the Standard Deviation, Properties of the Standard Deviation, Sheppard's Correction for Variance, Empirical Relations Between Measures of Dispersion, Absolute and Relative Dispersion; Coefficient of Variation, Standardized Variable; Standard Scores, Software and Measures of Dispersion. • **Introduction to R:** Basic syntax, data types, variables, operators, control statements, R-functions, R –Vectors, R – lists, R Arrays.	
Unit II	• **Moments, Skewness, and Kurtosis** : Moments , Moments for Grouped Data, Relations Between Moments , Computation of Moments for Grouped Data, Charlie's Check and Sheppard's Corrections, Moments in Dimensionless Form, Skewness, Kurtosis, Population Moments, Skewness, and Kurtosis. • **Elementary Probability Theory**: Definitions of Probability, Conditional Probability; Independent and Dependent Events, Mutually Exclusive Events, Mathematical Expectation, Relation Between Population, Sample Mean, and Variance, Combinations, Sterling's Approximation to n!. • **Elementary Sampling Theory**: Sampling Theory, Random Samples and Random Numbers, Sampling With and Without Replacement, Sampling Distributions, Sampling Distribution of Means, Sampling Distribution of Proportions, Sampling Distributions of Differences and Sums, Software Demonstration of Elementary Sampling Theory.	15 Lectures
Unit III	• **Statistical Estimation Theory**: Estimation of Parameters, Unbiased Estimates, Efficient Estimates, Point Estimates and Interval Estimates, Their Reliability, Confidence Interval Estimates of Population Parameters, Probable Error. • **Statistical Decision Theory:** Statistical Decisions, Statistical Hypotheses, Tests of Hypotheses and Significance, or Decision Rules, Type I and Type II Errors, Level of Significance, Tests Involving Normal Distributions, Two-Tailed and One-Tailed Tests, Special Tests, Operating- Characteristic Curves; the Power of a Test, p-Values for Hypotheses Tests • **Statistics in R:** mean, median, mode, Normal Distribution, Binomial Distribution, Frequency Distribution in R.	15 Lectures

Course Name: Computer Oriented Statistical Techniques Practical **Credits: 1 Type : Practical**	
Expected Course Outcomes On completion of this course, students will be able to 1.Apply measures of central tendencies to vectors, data frame and excel data using R 2.Set and Test Hypothesis and take statistical decisions.	
Practical No.	Title
1	Mean, Median, Mode
2	Variance, Standard Deviation, Covariance, Correlation
3	Range, Quartile, Interquartile Range, Histogram
4	Skewness, Kurtosis
5	Hypothesis Testing
6	Normal and Binomial Distribution

References

1. Statistics, By Murray.R.Spiegel, Larry J. Stephens, Published by McGraw –Hill International, 4th Edition.
2. Fundamental of Mathematical Statistics, By S.C. Gupta and V.K. Kapoor, Published By Sultan Chand and Sons, 11th Edition, 2011

Scheme of Evaluation:

I) Continuous Internal Evaluation (25 Marks)	
Class Test	10 Marks
Assignment/ Project and Viva/ Presentation	15 Marks
II) Theory Examination (50 Marks)	
Semester End Examination based on entire syllabus	50 Marks
III) Practical Examination (25 Marks)	
Certified Journal	5 marks
Viva Voce	5 marks
Practical exam	15 marks