



# JAI HIND COLLEGE BASANTSING INSTITUTE OF SCIENCE & J.T.LALVANI COLLEGE OF COMMERCE

# (AUTONOMOUS) "A" Road, Churchgate, Mumbai - 400 020, India.

Affiliated to University of Mumbai

Program : B.Sc.

**Proposed Subject : Mathematics (Applied Component)** Python and R Programming – II

Semester VI

Credit Based Semester and Grading System (CBGS) with effect from the academic year 2020-21

## T.Y.B.Sc. (Applied Component-Mathematics) Python and R Programming – II Syllabus

| Academic year 2020-2021 |  |         |                   |
|-------------------------|--|---------|-------------------|
| Semester VI             |  |         |                   |
| Course<br>Code          | Course Title                               | Credits | Lectures<br>/Week |
| SMAT605AC               | Python and R Programming - II              | 2.5     | 4                 |
| SMAT605ACPR             | Practical of Python and R Programming - II | 2.5     | 4                 |



## **Semester VI – Theory**

| SMAT<br>605AC         Objectives: Students will try to learn:         • Write Python functions to facilitate code reuse         • Use Python to read and write files         • Make their code robust by handling errors and exceptions properly         • Work with the Python standard library         • Explore Python's object-oriented features         • Search text using regular expressions         Outcomes: Students will be able to:         • Understand the concepts of object-oriented programming as used in Python: classes, subclasses, in - heritance, and overriding.         • Develop function based program.         • To understand and apply problem-solving skills using syntactically simple language.         Unit I       Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.         (b) Modules: Creating and importing own module       (c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.         Unit II       Object-oriented programming and Regular Expressions in Python       15 L         (a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in muthon       15 L | Course:      | Python and R Programming- II (Credits : 2.5, Lectures/Week:  | 4)    |
|--|--------------|--|-------|
| Objectives: Students will try to learn:         • Write Python functions to facilitate code reuse         • Use Python to read and write files         • Make their code robust by handling errors and exceptions properly         • Work with the Python standard library         • Explore Python's object-oriented features.         • Search text using regular expressions         Outcomes: Students will be able to:         • Understand the concepts of object-oriented programming as used in Python: classes, subclasses, in- heritance, and overriding.         • Develop function based program.         • To understand and apply problem-solving skills using syntactically simple language.         Unit I       Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.         (b) Modules: Creating and importing own module       (c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.         Unit II       Object-oriented programming and Regular Expressions in Python       15 L         (a) Object-oriented programming and Regular Expressions in Objects in inheritance, abstraction, polymorphism, Classes and Objects in rothon       15 L          | SMAT 605 A C |  |       |
| <ul> <li>Write Python functions to facilitate code reuse</li> <li>Use Python to read and write files</li> <li>Make their code robust by handling errors and exceptions properly</li> <li>Work with the Python standard library</li> <li>Explore Python's object-oriented features</li> <li>Search text using regular expressions</li> <li>Outcomes: Students will be able to:</li> <li>Understand the concepts of object-oriented programming as used in Python: classes, subclasses, in- heritance, and overriding.</li> <li>Develop function based program.</li> <li>To understand and apply problem-solving skills using syntactically simple language.</li> <li>Unit I Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> <li>Unit II Object-oriented programming and Regular Expressions in Python</li> <li>15 L</li> <li>(a) Object-oriented programming and Regular Expressions in Python</li> <li>15 L</li> </ul>   | Objectiv     | es: Students will try to learn:  |       |
| <ul> <li>Write Python functions to facilitate code reuse</li> <li>Use Python to read and write files</li> <li>Make their code robust by handling errors and exceptions properly</li> <li>Work with the Python standard library</li> <li>Explore Python's object-oriented features</li> <li>Search text using regular expressions</li> </ul> Outcomes: Students will be able to: <ul> <li>Understand the concepts of object-oriented programming as used in Python: classes, subclasses, in- heritance, and overriding.</li> <li>Develop function based program.</li> <li>To understand and apply problem-solving skills using syntactically simple language.</li> </ul> Unit I <ul> <li>Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications. Unit II Object-oriented programming and Regular Expressions in Python 15 L (a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in putbon</li></ul>   | Objectiv     |  |       |
| <ul> <li>Use Python to read and write files</li> <li>Make their code robust by handling errors and exceptions properly</li> <li>Work with the Python standard library</li> <li>Explore Python's object-oriented features</li> <li>Search text using regular expressions</li> </ul> Outcomes: Students will be able to: <ul> <li>Understand the concepts of object-oriented programming as used in Python: classes, subclasses, in-heritance, and overriding.</li> <li>Develop function based program.</li> <li>To understand and apply problem-solving skills using syntactically simple language.</li> </ul> Unit I Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables. <ul> <li>(h) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications. Unit II Object-oriented programming and Regular Expressions in Python <ul> <li>15 L</li> </ul></li></ul>   | • Write      | Python functions to facilitate code reuse  |       |
| <ul> <li>Make their code robust by handling errors and exceptions properly</li> <li>Work with the Python standard library</li> <li>Explore Python's object-oriented features</li> <li>Search text using regular expressions</li> </ul> Outcomes: Students will be able to: <ul> <li>Understand the concepts of object-oriented programming as used in Python: classes, subclasses, in-heritance, and overriding.</li> <li>Develop function based program.</li> <li>To understand and apply problem-solving skills using syntactically simple language.</li> </ul> Unit I Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables. <ul> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications. Unit II Object-oriented programming and Regular Expressions in Python <ul> <li>15 L</li> </ul></li></ul>   | • Use P      | ython to read and write files  |       |
| <ul> <li>Work with the Python standard library</li> <li>Explore Python's object-oriented features</li> <li>Search text using regular expressions</li> <li>Outcomes: Students will be able to:         <ul> <li>Understand the concepts of object-oriented programming as used in Python: classes, subclasses, in- heritance, and overriding.</li> <li>Develop function based program.</li> <li>To understand and apply problem-solving skills using syntactically simple language.</li> </ul> </li> <li>Unit I Functions And Modules in Python         <ul> <li>15 L</li> <li>(a) Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> </ul> <li>Unit II Object-oriented programming and Regular Expressions in Python         <ul> <li>(a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in puthon</li> </ul> </li> </li></ul>  | • Make       | their code robust by handling errors and exceptions properly   |       |
| <ul> <li>Explore Python's object-oriented features</li> <li>Search text using regular expressions</li> <li>Outcomes: Students will be able to:         <ul> <li>Understand the concepts of object-oriented programming as used in Python: classes, subclasses, in- heritance, and overriding.</li> <li>Develop function based program.</li> <li>To understand and apply problem-solving skills using syntactically simple language.</li> </ul> </li> <li>Unit I Functions And Modules in Python 15 L         <ul> <li>(a) Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> </ul> <li>Unit II Object-oriented programming and Regular Expressions in Python 15 L         <ul> <li>(a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in puthon</li> </ul> </li> </li></ul>   | • Work       | with the Python standard library   |       |
| <ul> <li>Search text using regular expressions</li> <li>Outcomes: Students will be able to:         <ul> <li>Understand the concepts of object-oriented programming as used in Python: classes, subclasses, in- heritance, and overriding.</li> <li>Develop function based program.</li> <li>To understand and apply problem-solving skills using syntactically simple language.</li> </ul> </li> <li>Unit I Functions And Modules in Python 15 L         <ul> <li>(a) Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> </ul> <li>Unit II Object-oriented programming and Regular Expressions in Python 15 L         <ul> <li>(a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in puthon</li> </ul> </li> </li></ul>  | • Explo      | re Python's object-oriented features   |       |
| Outcomes: Students will be able to:         • Understand the concepts of object-oriented programming as used in Python: classes, subclasses, in- heritance, and overriding.         • Develop function based program.         • To understand and apply problem-solving skills using syntactically simple language.         Unit I       Functions And Modules in Python         (a) Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.       (b) Modules: Creating and importing own module         (c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.       15 L         Unit II       Object-oriented programming and Regular Expressions in Python       15 L         (a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in puthon       15 L  | • Search     | n text using regular expressions   |       |
| <ul> <li>Understand the concepts of object-oriented programming as used in Python: classes, subclasses, in- heritance, and overriding.</li> <li>Develop function based program.</li> <li>To understand and apply problem-solving skills using syntactically simple language.</li> <li>Unit I Functions And Modules in Python 15 L</li> <li>(a) Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> <li>Unit II Object-oriented programming and Regular Expressions in Python 15 L</li> <li>(a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in putbon</li> </ul>   | Outcom       | se: Students will be able to:  |       |
| <ul> <li>Understand the concepts of object-oriented programming as used in Python: classes, subclasses, in- heritance, and overriding.</li> <li>Develop function based program.</li> <li>To understand and apply problem-solving skills using syntactically simple language.</li> <li>Unit I Functions And Modules in Python 15 L         <ul> <li>(a) Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> </ul> </li> <li>Unit II Object-oriented programming and Regular Expressions in Python 15 L         <ul> <li>(a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in puthon</li> </ul> </li> </ul>   | Outcom       | s. students will be able to.   |       |
| <ul> <li>Subclasses, in- heritance, and overriding.</li> <li>Develop function based program.</li> <li>To understand and apply problem-solving skills using syntactically simple language.</li> <li>Unit I Functions And Modules in Python 15 L</li> <li>(a) Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> <li>Unit II Object-oriented programming and Regular Expressions in Python 15 L</li> <li>(a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in python</li> </ul>  | • Under      | stand the concepts of object-oriented programming as used in Python: class   | ses   |
| <ul> <li>Develop function based program.</li> <li>To understand and apply problem-solving skills using syntactically simple language.</li> <li>Unit I Functions And Modules in Python 15 L         <ul> <li>(a) Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> </ul> </li> <li>Unit II Object-oriented programming and Regular Expressions in Python 15 L         <ul> <li>(a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in puthon</li> </ul> </li> </ul>  | subcla       | usses, in-heritance, and overriding.   | ,005, |
| <ul> <li>To understand and apply problem-solving skills using syntactically simple language.</li> <li>Unit I Functions And Modules in Python 15 L         <ul> <li>(a) Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> </ul> </li> <li>Unit II Object-oriented programming and Regular Expressions in Python 15 L         <ul> <li>(a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in python</li> </ul> </li> </ul>   | • Devel      | op function based program.   |       |
| Unit I       Functions And Modules in Python       15 L         (a) Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.       15 L         (b) Modules: Creating and importing own module       (c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.       15 L         Unit II       Object-oriented programming and Regular Expressions in Python       15 L         (a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in python       15 L  | • To un      | derstand and apply problem-solving skills using syntactically simple langua  | ige.  |
| Unit IFunctions And Modules in Python15 L(a) Functions: Definition, Advantages of functions, function parameters,<br>formal parameters, actual parameters, global and local variables.<br>(b) Modules: Creating and importing own module<br>(c) Python File Input-Output: Opening and closing files, various types<br>   |              |  | C     |
| <ul> <li>(a) Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> <li>Unit II Object-oriented programming and Regular Expressions in Python</li> <li>15 L</li> <li>(a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in python</li> </ul>  | Unit I       | Functions And Modules in Python  | 15 L  |
| <ul> <li>(a) Functions: Definition, Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> <li>Unit II Object-oriented programming and Regular Expressions in Python 15 L</li> <li>(a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in python</li> </ul>   |              | 3.3 X 27 AN A 1 F 1  |       |
| <ul> <li>(a) Tunctions Dominion, no valuages of functions, function parameters, formal parameters, actual parameters, global and local variables.</li> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> <li>Unit II Object-oriented programming and Regular Expressions in Python 15 L</li> <li>(a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in python</li> </ul>   |              | (a) Functions: Definition, Advantages of functions, function parameters  |       |
| <ul> <li>(b) Modules: Creating and importing own module</li> <li>(c) Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.</li> <li>Unit II Object-oriented programming and Regular Expressions in Python 15 L</li> <li>(a) Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in python</li> </ul>   |              | formal parameters, actual parameters, global and local variables.  |       |
| (c) Python File Input-Output: Opening and closing files, various types<br>of file modes, reading and writing to files, manipulating directories.<br>Iterables, iterators and their problem solving applications.15 LUnit IIObject-oriented programming and Regular Expressions in Python<br>inheritance, abstraction, polymorphism, Classes and Objects in<br>python15 L   |              | (b) Modules: Creating and importing own module   |       |
| of file modes, reading and writing to files, manipulating directories.<br>Iterables, iterators and their problem solving applications.15 LUnit IIObject-oriented programming and Regular Expressions in Python<br>inheritance, abstraction, polymorphism, Classes and Objects in<br>python15 L   |              | (c) Python File Input-Output: Opening and closing files, various types   |       |
| Iterables, iterators and their problem solving applications.         Unit II       Object-oriented programming and Regular Expressions in Python       15 L         (a)       Object-oriented programming concepts: encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in python       15 L  |              | of file modes, reading and writing to files, manipulating directories.   |       |
| Unit II       Object-oriented programming and Regular Expressions in Python       15 L         (a) Object-oriented programming concepts:       encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in python  |              | Iterables, iterators and their problem solving applications.   |       |
| Unit IIObject-oriented programming and Regular Expressions in Python15 L(a)Object-oriented programming concepts: encapsulation,<br>inheritance, abstraction, polymorphism, Classes and Objects in<br>python15 L  |              |  |       |
| (a) <b>Object-oriented programming concepts:</b> encapsulation, inheritance, abstraction, polymorphism, Classes and Objects in python  | Unit II      | <b>Object-oriented programming and Regular Expressions in Python</b>   | 15 L  |
| (a) <b>Object-oriented programming concepts:</b> encapsulation,<br>inheritance, abstraction, polymorphism, Classes and Objects in<br>python  |              |  |       |
| inheritance, abstraction, polymorphism, Classes and Objects in python  |              | (a) <b>Object-oriented programming concepts:</b> encapsulation,  |       |
| python   |              | inheritance, abstraction, polymorphism, Classes and Objects in   |       |
| pytion   |              | python   |       |
| (b) Exception handling: Keywords to handle exceptions such try, catch,   |              | (b) Exception handling: Keywords to handle exceptions such try, catch,   |       |
| except, else, finally, raise.  |              | except, else, finally, raise.  |       |
| (c) Regular Expressions: Concept of regular expression, various types  |              | (c) Regular Expressions: Concept of regular expression, various types  |       |
| of regular expressions, using match function.  |              | of regular expressions, using match function.  |       |
|  |              |  |       |
| Unit IIIPlots and Descriptive Statistics using R Programming15 L   | Unit III     | Plots and Descriptive Statistics using R Programming   | 15 L  |
| (a) Don Diot Histogram Die Chart Der Diet Diet Frantier 2D Diet  |              | (a) Day Diot Histogram Die Chart Day Diet Diet Expedier 2D Diet  |       |
| (a) Dat FIOL, FISCOGIAN, FIE CHARL, BOX FIOL, FIOL FUNCTION, 5D FIOL.<br>(b) Descriptive Statistics: Measures of central tendency. Measures of   |              | (a) Dat FIOL, FISCOGIAIII, FIE CHAIL, BOX FIOL, FIOL FUNCTION, 3D FIOL.<br>(b) Descriptive Statistics: Measures of central tendency. Measures of |       |
| variability Correlation  |              | variability Correlation  |       |

| Unit IV             | Regression using R Programming   | 15 L     |
|---------------------|--|----------|
|                     | (a) Linear Regression and application  |          |
|                     | (b) Multiple regression  |          |
|                     | (c) Logistic Regression  |          |
| Reference           | ces:   |          |
| • Practient et al., | cal Programming: An Introduction to Computer Science Using Python, Pau<br>Pragmatic Bookshelf, 2nd Edition 2014. | l Gries, |
| • Begin 2008.       | ning Python: From Novice to Professional, Magnus Lie Hetland, Apress,2   | edition  |
| • Wickl<br>York.    | nam, H. & amp; Grolemund, G. (2018). for Data Science. O'Reilly: New Available for free at http://r4ds.had.co.nz |          |
| • Paul              | Tetor: R Cook Book, O'Reilly,  |          |
| http://             | www.bagualu.net/wordpress/wpcontent/uploads/2015/10/R_Cookbook.pdf   |          |
| Addition            | nal References:  |          |
| • Introd            | uction to Computer Science using Python, Charles Dierbach, Wiley, 2013.  |          |
| • The Q             | uick Python Book, Naomi Ceder, Manning Publications; Third Edition, 2018   | 3        |

 R for Dummies; 2 edition <u>http://sgpwe.izt.uam.mx/files/users/uami/gma/R\_for\_dummies.pdf</u> (2015)



## Semester VI – Practical

| Cou<br>SM<br>605<br>AC | irse:<br>AT<br>PR | Practical of Python and R Programming (Credits 2.5 : Practical/Week: 4) |
|------------------------|-------------------|---|
| 1.                     | Progra            | am based on Function and anonymous function.                            |
| 2.                     | Progra            | am based on Modules.  |
| 3.                     | Progra            | ams based on File processing.   |
| 4.                     | Progra            | ams based on Exception handling   |
| 5.                     | Progra            | ams based on regular expressions  |
| 6.                     | Progra            | ams based on Plotting a data  |
| 7.                     | Progra            | ams based on Descriptive Statistics                                     |
| 8.                     | Linea             | r Regression and application  |
| 9.                     | Multi             | ple regression  |
| 10.                    | Logis             | tic Regression  |
|                        |                   | Jinn W  |

## **Evaluation Scheme**

## **Evaluation scheme for Theory courses**

#### I. Continuous Assessment (C.A.) - 40 % - 40 Marks

| Sr. No. | Evaluation type  | Marks |
|---------|--|-------|
| 1.      | <b>C.AI</b> : It will be conducted either using any open source learning management system or by taking a test | 20    |
| 2.      | <b>C.AII</b> : Assignments / Project (maximum 5 students in a group)   | 20    |

### II. Semester End Examination (SEE)- 60 % - 60 Mark , Duration 2 Hrs

**Theory Question Paper Pattern:-**

| All Questions are Compulsory |                |          |       |
|------------------------------|----------------|----------|-------|
| Question                     | Options        | Based on | Marks |
| 1.                           | Any 3 out of 5 | Unit I   | 15    |
| 2.                           | Any 3 out of 5 | Unit II  | 15    |
| 3.                           | Any 3 out of 5 | Unit III | 15    |
| 4.                           | Any 3 out of 5 | Unit IV  | 15    |

# Evaluation scheme for Practical courses- 100 Marks

Each student will maintain an e-journal. After every practical, student will upload his practical in the form of documents along with the screen shots of output on any LMS.

## [1] CA of Practical Course - 40 % - 40 Marks:

Individual student or group of students (maximum 5 students) shows implementation of a given problem not from Practical List and its performance.

### [2] SEE of Practical Course - 60 % - 60 Marks:

| Sr. No. | Heading                       | Marks |
|---------|-------------------------------|-------|
| 1.      | Journal                       | 10    |
| 2.      | Implementation of Python Code | 25    |
| 3       | Implementation of R Code      | 25    |
|         | Total                         | 60    |