

**Nirma University  
Institute of Technology**

**Computer Science & Engineering Department**

**Course Policy Document**

**B.Tech. Computer Science and Engineering**

**Semester: IV Academic Year: 2023 Term: Even**

|                                       |   |   |
|---------------------------------------|---|---|
| <b><u>Course Code &amp; Name</u></b>  | : | 2CSOE78 : Programming for Scientific Computing  |
| <b>Credit Details</b>                 | : | 3 credit course : [2 0 2 3] ([L T P C])   |
| <b><u>Course Co ordinator</u></b>     | : | Dr Sapan H Mankad   |
| <b><u>Contact No. &amp; Email</u></b> | : | 07971652573<br>Email: <a href="mailto:sapanmankad@nirmauni.ac.in">sapanmankad@nirmauni.ac.in</a>                    |
| <b>Office</b>                         | : | New Building, 5 <sup>th</sup> Floor   |
| <b><u>Visiting Hours</u></b>          | : | Monday to Friday : 8:45 am to 4:00 pm<br>Saturday (Odd) : 8:45am to 4:45pm  |
| <b><u>Course Blog</u></b>             | : | -   |
| <b>Course Website</b>                 | : | <a href="https://lms.nirmauni.ac.in/course/view.php?id=1526">https://lms.nirmauni.ac.in/course/view.php?id=1526</a> |
| <b><u>Course Faculty</u></b>          | : | Dr Sapan H Mankad and Dr Usha Patel   |
| <b><u>Contact No. &amp; Email</u></b> | : | <a href="mailto:sapanmankad@nirmauni.ac.in">sapanmankad@nirmauni.ac.in</a>  |
| <b><u>Office</u></b>                  | : | N block faculty cabins  |
| <b><u>Visiting Hours</u></b>          | : | Monday to Friday : 8:45 am to 4:00 pm<br>Saturday (Odd) : 8:45am to 4:45pm  |

# **1. Introduction to Course**

## 1.1 Importance of the course

- Basically Python is a language that can be used for developing anything and everything you want.
- Now a days, most of the organization wants to develop data science and analytics to enhance decision making.
- Huge community support and large amount of API is available.

## 1.2 Course Objective:

- To understand why Python is a useful scripting language for developers.
- To learn how to design and program Python applications

## 1.3 Pre-requisite:

- Awareness of basics Programming knowledge.

# **2. Course Learning Outcomes**

After successful completion of this course, student will be able to:

1. write computational programs at a high level of abstraction
2. use standard programming constructs like repetition, selection, functions, composition, modules, aggregated data
3. implement and evaluate the results of scientific computing problems, using established program libraries

# **3. Syllabus**

**Unit I:** Introduction to Computational Science, Applications involving scientific computing, Tools and languages to solve complex scientific problems **Unit II:** Programming in Python- Interpreter and its environment; Introduction to data types, concepts of mutability, operators and variables; random numbers, user inputs, statements; branching, conditional and iteration; functions, file handling, error handling and exceptions

**Unit III:** Object-oriented programming, classes and methods - encapsulation, inheritance

**Unit IV:** Array computing and curve plotting, vectors and higher-dimensional arrays, matrices, numPy, sciPy and Matplotlib

**Unit V:** Python Pandas - Data alignment, aggregation, summarization, computation and analysis with Pandas

**Unit VI:** Scientific computation using python - Statistical data analysis, image processing, web development and hardware interfacing using Python

## **3.1 Self-study:**

Tools and languages to solve complex scientific problems and hardware interfacing using Python.

# **4 Laboratory details**

Laboratory work will be based on above syllabus with following experiments to be performed.

Each experiment will be of 10 marks. Evaluation for 100 marks will be done throughout the semester as part of the Continuous Evaluation scheme. The assessment of Laboratory work is as under:

| Total Marks | Continuous Evaluation |            |           | Semester End /Viva voce Evaluation |           |
|-------------|-----------------------|------------|-----------|------------------------------------|-----------|
|             | No. of Practicals     | Max. Marks | Weightage | Max Marks                          | Weightage |
| 100 marks   | 10                    | 100        | 75%       | 25                                 | 25%       |

### **Assessment Policy for Lab Practicals:**

Assessment of laboratory work (practical) will be done base of following criteria.

- Code logic
- Design of User Interface
- Timely submission
- Regularity in attending lab sessions
- Error Solving

### **Tutorial details: ( problem sheet, schedule, assessment policy)**

NA

### **References:**

1. Hans Petter Langtangen, A Primer on Scientific Programming with Python (Link)
2. Claus Fuhrer, Jan Erik Solem, Olivier Verdier, Scientific Computing with Python 3, Packt Publishing Limited
3. Martin C. Brown, Python: The Complete Reference, McGraw Hill Education
4. Hemant Kumar Mehta, Mastering Python Scientific Computing, Packt Publishing Limited
5. Sergio J. Rojas G., Erik A. Christensen, Francisco J. Blanco-Silva, Learning SciPy for Numerical and Scientific Computing, Packt Publishing Limited

## List of Practicals

| Sr. No.          | Week No.#                            | List of Experiments  | Mapped CLO       |         |      |                  |            |                |      |              |        |                                      |          |                                 |   |
|------------------|--------------------------------------|--|------------------|---------|------|------------------|------------|----------------|------|--------------|--------|--------------------------------------|----------|---------------------------------|---|
| 1                | 1,2,3                                | <div>1. Write a function areaTriangle that takes the lengths of three sides of the triangle as input parameters and returns the area of the triangle as an output. Also, assert that the sum of the length of any two sides is higher than the third side.</div> <div>2. Write a python program for three different scientific formulas of your own choice.</div> <div>3. Write a function that takes two numbers as input parameters and returns True or False depending on whether they are co-primes. Two numbers are said to be co-prime if they do not have any common divisor other than one.</div>  | 1,2              |         |      |                  |            |                |      |              |        |                                      |          |                                 |   |
| 2                | 4                                    | <div>1. Write a function that takes a string as a parameter and returns a string with every successive repetitive character replaced with a star(*). For Example, 'balloon' is returned as 'bal*o*n'. (hint: <a href="https://www.w3schools.com/python/python_strings.asp">https://www.w3schools.com/python/python_strings.asp</a>)</div> <div>2. Write a function that takes a number as n input parameter and returns the corresponding text in words; for example, on input 452, the function should return 'Four Five Two'. Use an indexed list for mapping to digits to their string representation.</div> <div>3. Write a recursive function that takes x value as an input parameter and print x-digit strictly in increasing number. [i.e. x = 6 than output 67891011]</div> | 1,2              |         |      |                  |            |                |      |              |        |                                      |          |                                 |   |
| 3                | 5                                    | <div>Write a python program that reads the contents of the file poem.txt and count the number of alphabets blank spaces lowercase letters and uppercase letters the number of words starting from vowel and the number of occurrences of each word in the file (For file handling, refer: <a href="https://www.w3schools.com/python/python_file_open.asp">https://www.w3schools.com/python/python_file_open.asp</a>)</div> <div>Important links</div> <div><a href="https://buomsoo-kim.github.io/colab/2018/04/15/Colab-Importing-CSV-and-JSON-files-in-Google-Colab.md/">https://buomsoo-kim.github.io/colab/2018/04/15/Colab-Importing-CSV-and-JSON-files-in-Google-Colab.md/</a>)</div>  | 1,2              |         |      |                  |            |                |      |              |        |                                      |          |                                 |   |
| 4                | 6,7                                  | <div>1. Define a class Bank that keeps track of bank customers. The class should contain the following data member:</div> <table><tr><th>Data member name</th><th>Details</th></tr><tr><td>Name</td><td>Name of customer</td></tr><tr><td>accountNum</td><td>Account Number</td></tr><tr><td>Type</td><td>Account Type</td></tr><tr><td>Amount</td><td>Amount deposited in the bank account</td></tr><tr><td>Interest</td><td>Interest earned by the customer</td></tr></table> <div>The class should support the following methods:</div> <div><ul style="list-style-type: none"><li>__init__ for initializing the data members.</li></ul></div>  | Data member name | Details | Name | Name of customer | accountNum | Account Number | Type | Account Type | Amount | Amount deposited in the bank account | Interest | Interest earned by the customer | 1 |
| Data member name | Details                              |  |                  |         |      |                  |            |                |      |              |        |                                      |          |                                 |   |
| Name             | Name of customer                     |  |                  |         |      |                  |            |                |      |              |        |                                      |          |                                 |   |
| accountNum       | Account Number                       |  |                  |         |      |                  |            |                |      |              |        |                                      |          |                                 |   |
| Type             | Account Type                         |  |                  |         |      |                  |            |                |      |              |        |                                      |          |                                 |   |
| Amount           | Amount deposited in the bank account |  |                  |         |      |                  |            |                |      |              |        |                                      |          |                                 |   |
| Interest         | Interest earned by the customer      |  |                  |         |      |                  |            |                |      |              |        |                                      |          |                                 |   |

|                            |                        | <ul style="list-style-type: none"><li>• deposit for depositing money in the members.</li><li>• withdrawal for withdrawing money from the account</li><li>• findInterest that determines the interest on the basis of amount in the account</li></ul> <table><tr><th>Amount</th><th>Interest per annum (%)</th></tr><tr><td>&gt;= 5,00,000</td><td>8</td></tr><tr><td>&gt;= 3,00,000 and &lt; 5,00,000</td><td>7</td></tr><tr><td>&gt;=1,00,000 and &lt;3,00,000</td><td>5</td></tr><tr><td>&lt;1,00,000</td><td>3</td></tr></table> <p>2. Define a base class Person, having attributes name, birthdate and city. Define the class Student that derives from Person class which is having attributes like rollno, branch, totalMarks and year as data member. The class should contain the instance method __init__ and the abstract method percentage. Define two classes Grad and PostGrad which inherit from the base class Student. Both the classes should define their __init__ method which asks user to enter totalMarks value and should override the abstract method percentage of the superclass. Note that totalMarks obtained are out of 600 and 400 for Grad and PostGrad classes respectively.</p>  | Amount | Interest per annum (%) | >= 5,00,000 | 8 | >= 3,00,000 and < 5,00,000 | 7 | >=1,00,000 and <3,00,000 | 5 | <1,00,000 | 3 |  |
|----------------------------|------------------------|--|--------|------------------------|-------------|---|----------------------------|---|--------------------------|---|-----------|---|--|
| Amount                     | Interest per annum (%) |  |        |                        |             |   |                            |   |                          |   |           |   |  |
| >= 5,00,000                | 8                      |  |        |                        |             |   |                            |   |                          |   |           |   |  |
| >= 3,00,000 and < 5,00,000 | 7                      |  |        |                        |             |   |                            |   |                          |   |           |   |  |
| >=1,00,000 and <3,00,000   | 5                      |  |        |                        |             |   |                            |   |                          |   |           |   |  |
| <1,00,000                  | 3                      |  |        |                        |             |   |                            |   |                          |   |           |   |  |
| 5                          | 8,9                    | <p>1. A magic square is a matrix all of whose row sums, column sums and the sums of the two diagonals are the same. (One diagonal of a matrix goes from the top left to the bottom right, the other diagonal goes from top right to bottom left.) Show by direct computation that if the matrix A is given by</p> <pre>A=np.array([[17, 24, 1, 8, 15],<br/>            [23, 5, 7, 14, 16],<br/>            [ 4, 6, 13, 20, 22],<br/>            [10, 12, 19, 21, 3],<br/>            [11, 18, 25, 2, 9]])</pre> <p>The matrix A has 5 row sums (one for each row), 5 column sums (one for each column) and two diagonal sums. These 12 sums should all be exactly the same, and you could verify that they are the same by printing them and "seeing" that they are the same. It is easy to miss small differences among so many numbers, though. Instead, verify that A is a magic square by constructing the 5 column sums and computing the maximum and minimum values of the column sums. Do the same for the 5 row sums, and compute the two diagonal sums. Check that these six values are the same. If the maximum and minimum values are the same, the flyswatter principle says that all values are the same.</p> <p>2. Create scientific calculator using numpy.</p> | 2,3    |                        |             |   |                            |   |                          |   |           |   |  |

|                |       |   |                |     |     |     |     |     |     |      |     |     |      |       |   |    |    |   |    |   |    |    |    |    |         |    |    |    |    |   |   |     |    |    |    |     |
|----------------|-------|---|----------------|-----|-----|-----|-----|-----|-----|------|-----|-----|------|-------|---|----|----|---|----|---|----|----|----|----|---------|----|----|----|----|---|---|-----|----|----|----|-----|
| 6              | 10    | <p>Plot a line graph that shows the runs scored between two-wicket fall in a one-day international match between India and England which is generated randomly. Example:</p> <table><tr><td>Fall of Wicket</td><td>1st</td><td>2nd</td><td>3rd</td><td>4th</td><td>5th</td><td>6th</td><td>7th</td><td>8th</td><td>9th</td><td>10th</td></tr><tr><td>India</td><td>5</td><td>35</td><td>24</td><td>0</td><td>99</td><td>1</td><td>35</td><td>15</td><td>27</td><td>14</td></tr><tr><td>England</td><td>10</td><td>55</td><td>34</td><td>21</td><td>2</td><td>7</td><td>118</td><td>29</td><td>32</td><td>10</td></tr></table> | Fall of Wicket | 1st | 2nd | 3rd | 4th | 5th | 6th | 7th  | 8th | 9th | 10th | India | 5 | 35 | 24 | 0 | 99 | 1 | 35 | 15 | 27 | 14 | England | 10 | 55 | 34 | 21 | 2 | 7 | 118 | 29 | 32 | 10 | 2,3 |
| Fall of Wicket | 1st   | 2nd   | 3rd            | 4th | 5th | 6th | 7th | 8th | 9th | 10th |     |     |      |       |   |    |    |   |    |   |    |    |    |    |         |    |    |    |    |   |   |     |    |    |    |     |
| India          | 5     | 35  | 24             | 0   | 99  | 1   | 35  | 15  | 27  | 14   |     |     |      |       |   |    |    |   |    |   |    |    |    |    |         |    |    |    |    |   |   |     |    |    |    |     |
| England        | 10    | 55  | 34             | 21  | 2   | 7   | 118 | 29  | 32  | 10   |     |     |      |       |   |    |    |   |    |   |    |    |    |    |         |    |    |    |    |   |   |     |    |    |    |     |
| 7              | 11    | Develop a python program that reads the data from a given CSV file, which is having phone usage data using a different branded mobile phone. Determine if the usage patterns for users differ between different devices. For example, do users using Samsung devices use more call minutes than those using LG devices?   | 2,3            |     |     |     |     |     |     |      |     |     |      |       |   |    |    |   |    |   |    |    |    |    |         |    |    |    |    |   |   |     |    |    |    |     |
| 8              | 12,13 | <div>1. Design a python program which performs the linear regression operation on the given data to predict the house price.</div> <div>2. Design a python program that generates the 100 random variables and computes the mean, median and mode for the same.</div>   | 3              |     |     |     |     |     |     |      |     |     |      |       |   |    |    |   |    |   |    |    |    |    |         |    |    |    |    |   |   |     |    |    |    |     |
| 9              | 14    | Develop a python program that reads the image, display matrix representation of an image creates a histogram of the image and apply the smoothing effect on an image.   | 3              |     |     |     |     |     |     |      |     |     |      |       |   |    |    |   |    |   |    |    |    |    |         |    |    |    |    |   |   |     |    |    |    |     |
| 10.            | 15    | Develop a python program which takes the video as an argument and extract all the frames from a video. Select specific frames and recreate the video, which has selected frames only.   | 3              |     |     |     |     |     |     |      |     |     |      |       |   |    |    |   |    |   |    |    |    |    |         |    |    |    |    |   |   |     |    |    |    |     |
| 11             | Extra | <b>Challenging practical list (Optional)</b> <div>1. Write a python program to perform a binary search tree algorithm.</div> <div>2. Develop a python program for checking the typing speed test and show the accuracy along with word per minute.</div> <div>3. Create a Sudoku game using python</div>  | 3              |     |     |     |     |     |     |      |     |     |      |       |   |    |    |   |    |   |    |    |    |    |         |    |    |    |    |   |   |     |    |    |    |     |

## 6. Course Evaluation Methodology

### 6.1 Component wise Continuous Evaluation & Semester End Examination weightage:

| COMPONENTS WEIGHTAGE |                            |     |
|----------------------|----------------------------|-----|
| Course Details       | Continuous Evaluation (CE) | LPW |

| Subject Code | Subject Name                                     | Course Coordinator | No of Assignments | Class Test | Sessional Exam | Special Assignment */Tutorials |                 |     | No. of Practicals | Lab Work | Term End Exam |
|--------------|--|--------------------|-------------------|------------|----------------|--------------------------------|-----------------|-----|-------------------|----------|---------------|
|              |  |                    |                   | WT         | WT             | Type of Assignment             | No of Tutorials | WT  |                   | WT       | WT            |
|              | 2COE78<br>Programming for Scientific Programming | Dr Sapan H Mankad  | 01                | 0.35       | 0.35           | Project                        | -               | 0.3 | 10                | 0.75     | 0.25          |

## 6.2 Assessment Policy for Continuous Evaluation (CE)

The marks divisions for the continuous evaluation will be as follow:

- Class Test (35 Marks)
- Sessional Exam (35 Marks)
- Special assignment (30 Marks)

## 6.3 Assessment Policy for Laboratory (LPW)

Assessment of Laboratory and Project Work comprises of two components.

- Continuous assessment for laboratory experiments will be conducted. There will be 10 experiments, each carrying weightage of 10 marks. At the end of the course total marks obtained out of 100 will be converted according to weightage assigned. Assessment of Experiment will be carried out based on parameters like Code logic, Design of User Interface, Timely submission, Regularity in attending lab sessions, Error Solving, etc. during the session and the parameters mentioned in Section 4 of this document.
- A Viva voce examination for LPW component will be conducted as per academic calendar. It will carry a weightage of 25 marks.

## 6. 4 Assessment Policy for Semester End Examination (SEE)

Will be updated soon

## **7. Lesson Plan**

| <b>Lecture No.</b> | <b>Topic</b>   | <b>Mappe<br/>d<br/>CLO</b> |
|--------------------|--|----------------------------|
| 1                  | Introduction to Computational Science, Applications involving scientific computing, Tools and languages to solve complex scientific problems | 1                          |
| 2                  | Programming in Python- Interpreter and its environment; Introduction to data types   | 1                          |
| 3                  | Concepts of mutability   | 2                          |
| 4                  | Concepts of mutability   | 2                          |
| 5                  | Operators and variables; random numbers  | 1                          |
| 6                  | User inputs, statements; branching, conditional and iteration  | 1                          |
| 7                  | Functions  | 1                          |
| 8                  | File handling  | 1                          |
| 9                  | Error handling and exceptions  | 1                          |
| 10                 | Object-oriented programming, classes and methods   | 1                          |
| 11                 | Encapsulation  | 1                          |
| 12                 | Inheritance  | 1                          |
| 13                 | Array computing and curve plotting, vectors and higher dimensional arrays  | 2                          |
| 14                 | Matrices   | 2                          |
| 15                 | NumPy  | 3                          |
| 16                 | NumPy  | 3                          |
| 17                 | SciPy and Matplotlib   | 3                          |
| 18                 | Matplotlib   | 3                          |
| 19                 | Python Pandas - Data alignment, aggregation  | 2, 3                       |
| 20                 | Python Pandas Summarization  | 2, 3                       |
| 21                 | Python Pandas Summarization  | 2, 3                       |
| 22                 | Computation and analysis with Pandas   | 2, 3                       |
| 23                 | Computation and analysis with Pandas   | 2, 3                       |
| 24                 | Scientific computation using python - Statistical data analysis  | 3                          |



|    |   |   |
|----|---|---|
| 25 | Scientific computation using python - Statistical data analysis | 3 |
| 26 | Image processing  | 3 |
| 27 | Image processing  | 3 |
| 28 | Web development   | 3 |
| 29 | Web development   | 3 |
| 30 | Hardware interfacing using Python                               | 3 |

**Class Test:** As per academic calendar

**Sessional Exam:** As per academic calendar

### **Tutorial Plan**

| Tutorial No | Topic | Mapped CLO |
|-------------|-------|------------|
| NA          |       |            |

### **Course Assessment Schemes**

| Assessment scheme   | CE               |                       |   | LPW                          |                  | SEE          |
|---------------------|------------------|-----------------------|---|------------------------------|------------------|--------------|
| Component weightage | 0.4              |                       |   | 0.2                          |                  | 0.4          |
|                     | Quiz I&II<br>35% | Sessional Exam<br>35% | Term Paper/<br>Innovative Assignment<br>30% | Continuous Evaluation<br>75% | Viva Voce<br>25% | 3 hours exam |

### **Special/Innovative Assignments, Term Papers, mini Projects**

**etc.** • Type: Project

- Details: Students need to develop the project in group of 3 students. They need to design a solution for real time problem with GUI.
- Timeline: Submission date: 20<sup>th</sup> April 2023

## **9. Teaching-learning methodology**

- **Lectures:** Primarily Chalk and Black board will be used to conduct the course for off-line lectures. However, where required (also for on-line lectures), Power Point Presentations (PPTs), Video Lectures, Simulations / Animations, Program(s) Demonstration etc. will be used to enhance the teaching-learning process.
- **Laboratory:** Explanation of Experiment to be performed along with co-relation with theory will be given. At the end of each session assessment will be carried out based on parameters like completion of lab work that includes Logic Syntax, Error Solving, Timely submission of practical's and lab practical submission, Regularity in attending lab sessions, Discipline of student in the lab sessions /on-line sessions etc. Students will be quizzed to check their understanding of the experiment/exercise conducted.

## **10. Active learning techniques**

Following active learning techniques will be adopted for the course. •

The Muddiest Point: The technique consists of asking students to jot down a quick response to one question: "What was the muddiest point in [the lecture, discussion, homework assignment, film, etc.]" The term "muddiest" means "most unclear" or "most confusing."

## **11. Course Material**

Following course materials will be uploaded on the course website:

- Course Policy
- Lecture Notes (will be provided just before exam & not day to day) •
- Books / Reference Books / NPTEL video lectures
- Assignments, Tutorials, Lab Manuals
- Question bank (As and when topics are covered in class and lab sessions)
- Web-links, Video Lectures for reference
- Animations / Simulations, Softwares
- Advanced topics

## **12. Course Learning Outcome Attainment**

Following means will be used to assess attainment of course learning outcomes:

- Use of formal evaluation components of continuous evaluation, tutorials, laboratory work, semester end examination
- Informal feedback during course conduction

### **13. Academic Integrity Statement**

Students are expected to carry out assigned work under Continuous Evaluation (CE) component and LPW component independently. Copying in any form is not acceptable and will invite strict disciplinary action. Evaluation of corresponding component will be affected proportionately in such cases. Turnitin software will be used to check plagiarism wherever applicable. Academic integrity is expected from students in all components of course assessment.