

## Name of the Course: Advanced Python Programming

Sr. No.	Heading	Particulars
1	Description the course:	<p><b>Introduction:</b></p> <p>The Advanced Python Programming Course is designed to elevate coding skills to a more sophisticated level, offering participants a deeper understanding of Python's advanced features and capabilities. Building upon the foundations laid in basic Python courses, this advanced course delves into complex programming concepts and techniques.</p> <p><b>Relevance:</b></p> <p>As technology advances, the relevance of Python continues to grow. The Advanced Python Programming Course is a response to the increasing demand for skilled Python developers who can tackle intricate challenges in various domains, including software development, data science, artificial intelligence, and more.</p> <p><b>Usefulness:</b></p> <p>This course goes beyond basic syntax and introduces participants to advanced Python topics such as decorators, generators, metaclasses, and asynchronous programming. Learners gain valuable insights into optimizing code performance, enhancing code readability, and solving complex problems efficiently.</p> <p><b>Application:</b></p> <p>Graduates of this course can apply their advanced Python skills to tackle more complex programming tasks, develop scalable applications, and contribute to large-scale software projects. The course's emphasis on practical applications ensures that participants are well-equipped for real-world programming challenges.</p> <p><b>Interest:</b></p> <p>The course maintains an engaging learning experience, balancing theoretical concepts with hands-on projects that challenge participants to apply their knowledge creatively. This approach fosters a continued interest in Python programming and encourages learners to explore advanced topics with enthusiasm.</p> <p><b>Connection with Other Courses:</b></p> <p>The knowledge gained in the Advanced Python</p>

		<p>Programming Course establishes a strong foundation for further specialization in advanced Python libraries, frameworks, and application domains. This course acts as a bridge to more specialized fields such as machine learning, web development, and data engineering.</p> <p><b>Demand in the Industry:</b></p> <p>Professionals with advanced Python skills are highly sought after in the industry. The ability to leverage Python's advanced features for efficient problem-solving, code optimization, and system architecture places graduates of this course in high demand across diverse sectors.</p> <p><b>Job Prospects:</b></p> <p>Completing the Advanced Python Programming Course opens doors to advanced positions in software development, data engineering, scientific computing, and research. Job prospects include roles such as Python developer, data scientist, machine learning engineer, and backend developer, among others.</p>
2	<b>Vertical:</b>	SEC
3	<b>Type:</b>	Practical
4	<b>Credits:</b>	2 credits ( 1 credit = 15 Hours for Theory or 30 Hours of Practical work in a semester )
5	<b>Hours Allotted:</b>	60 Hours
6	<b>Marks Allotted:</b>	50 Marks
7	<p><b>Course Objectives(CO):</b></p> <p><b>CO 1.</b> Master OOPs principles, solving real-world problems.</p> <p><b>CO 2.</b> Create robust Python classes, transfer members efficiently.</p> <p><b>CO 3.</b> Understand and implement inheritance, utilize advanced polymorphism..</p> <p><b>CO 4.</b> Implement abstract classes, leverage interfaces for flexible code.</p> <p><b>CO 5.</b> Create and synchronize threads, mitigate deadlock issues.</p>	
8	<p><b>Course Outcomes (OC):</b></p> <p><b>OC 1.</b> Demonstrate comprehensive OOPs proficiency, apply principles effectively.</p> <p><b>OC 2.</b> Develop efficient, reusable classes, successfully transfer members.</p> <p><b>OC 3.</b> Ability to implement inheritance and apply advanced polymorphism.</p> <p><b>OC 4.</b> Ability to implement abstract classes, demonstrate flexibility through interfaces.</p> <p><b>OC 5.</b> Ability to thread creation, synchronization, and effective deadlock resolution.</p>	

9	<p><b>Modules:-</b></p> <p><b>Module 1 (30 hours):</b></p> <p><b>OOPs In Python:</b> Introduction to OOPs, Problems in Procedure Oriented Approach, Features of Object Oriented Programming System (OOPS), Constructors and Destructors,</p> <p><b>Classes and Objects-</b> Creating a Class, Self-Variable, Types of Variables, Types of Methods, Passing Members of One Class to Another Class</p> <p><b>Inheritance and Polymorphism:</b> Types of Inheritance, Constructors in Inheritance, Overriding Super Class Constructors and Methods, super() method, Polymorphism, Duck Typing , Operator Overloading, Method Overloading, Method Overriding</p> <p><b>Abstract Classes and Interfaces:</b> Abstract Class, Abstract Method, Interfaces in Python</p> <p><b>Threads in Python:</b> Creating Threads in Python, Single Tasking and Multitasking, Thread Synchronisation, Deadlock in Threads</p> <p><b>Module 2 (30 hours):</b></p> <p><b>Working with Databases:</b> DBMS, working with MySQL Database-retrieving, inserting, deleting, updating rows from table, Creating Database Tables through Python</p> <p><b>Exceptions:</b> Errors in a Python Program, Exceptions and Exceptions handling, User Defined Exceptions, Logging Exceptions,</p> <p><b>Networking:</b> TCP/IP Protocol Architecture, , User Datagram Protocol (UDP), FTP Architecture, Web Page Operations, Sending a Simple Mail</p> <p><b>Graphical User Interface:</b> Creating a GUI in Python, Widget classes, Layout Manager, Event Handling</p> <p><b>Data Science Tools:</b> Introduction to NumPy, Matplotlib, pandas, Scipy,</p>
10	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>1. Practical Programming: An Introduction to Computer Science Using Python 3, Paul Gries , Jennifer Campbell, Jason Montojo, Pragmatic Bookshelf, 2nd Edition, 2014</li> <li>2. Programming through Python, M. T Savaliya, R. K. Maurya&amp; G M Magar, Sybgen Learning India, 2020</li> </ol>
11	<p><b>Reference Books</b></p> <ol style="list-style-type: none"> <li>1. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018</li> <li>2. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017</li> <li>3. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed, 2018</li> </ol>

<b>12</b>	<b>Internal Continuous Assessment: 40%</b>	<b>Semester End Examination: 60%</b>												
<b>13</b>	<p>The internal evaluation will be determined by the completion of practical tasks and the submission of corresponding write-ups for each session. Each practical exercise holds a maximum value of 5 marks. The total evaluation, out of 50 marks, should be scaled down to a final score of 20 marks.</p> <hr/> <p><b>Total: 20 marks</b></p>	<p>A <b>Semester End Practical Examination</b> of <b>2 hours duration</b> for <b>30 marks</b> as per the paper pattern given below.</p> <p><b>Certified Journal</b> is <b>compulsory</b> for appearing at the time of Practical Exam</p> <hr/> <p><b>Total: 30 Marks</b></p>												
<b>14</b>	<p><b>Format of Question Paper:</b></p> <p><b>Total Marks: 30</b> <span style="float: right;"><b>Duration: 2 Hours</b></span></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Question</th><th style="width: 33%;">Practical Question Based On</th><th style="width: 33%;">Marks</th></tr> </thead> <tbody> <tr> <td><b>Q. 1</b></td><td>Module 1</td><td>12</td></tr> <tr> <td><b>Q. 2</b></td><td>Module 2</td><td>12</td></tr> <tr> <td><b>Q. 3</b></td><td>Viva</td><td>06</td></tr> </tbody> </table>		Question	Practical Question Based On	Marks	<b>Q. 1</b>	Module 1	12	<b>Q. 2</b>	Module 2	12	<b>Q. 3</b>	Viva	06
Question	Practical Question Based On	Marks												
<b>Q. 1</b>	Module 1	12												
<b>Q. 2</b>	Module 2	12												
<b>Q. 3</b>	Viva	06												