

DRONACHARYA

College of Engineering

Khentawas, Farrukh Nagar, Gurugram, Haryana

Approved by: All India Council for Technical Education (AICTE), New Delhi

Affiliated to: Maharshi Dayanand University (MDU) Rohtak, Haryana

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Academic Year 2022-23

Course Code/ Name: PCC-CSE-207G, Python Programming

Semester III

1. Project-Based Learning:

- **Purpose:** Apply Python skills to real-world projects.
- **Implementation:** Assign projects that require students to solve problems across different domains such as data analysis, web development, automation, or scientific computing. Encourage creativity and experimentation with Python libraries and frameworks (e.g., NumPy, Django, TensorFlow).

2. Interactive Coding Environments:

- **Purpose:** Provide hands-on coding experience in a collaborative setting.
- **Implementation:** Use online coding platforms (e.g., Jupyter Notebooks, Google Colab) where students can write and execute Python code in a browser environment. Integrate features for real-time collaboration, code sharing, and interactive visualizations (e.g., matplotlib, Plotly).

3. Gamification and Coding Challenges:

- **Purpose:** Foster engagement and problem-solving skills.
- **Implementation:** Organize coding competitions, hackathons, or coding challenges (e.g., Codeforces, LeetCode) where students can solve algorithmic problems using Python. Provide leaderboard rankings, feedback on coding efficiency, and opportunities for peer-to-peer learning.

4. Python for Data Science Workshops:

- **Purpose:** Introduce Python's role in data analysis and visualization.
- **Implementation:** Conduct workshops on Python libraries like pandas for data manipulation, matplotlib/seaborn for data visualization, and scikit-learn for

machine learning. Assign projects where students analyze datasets, build predictive models, and present insights using Python tools.

-

5. Version Control and Collaborative Development:

- **Purpose:** Teach industry-standard practices in software development.
- **Implementation:** Use version control systems like Git and platforms like GitHub or GitLab for collaborative Python project development. Teach branching, merging, and pull requests to facilitate teamwork and code review processes.

6. Integration with IoT and Hardware Projects:

- **Purpose:** Explore Python's role in controlling and interacting with hardware.
- **Implementation:** Introduce microcontroller platforms (e.g., Raspberry Pi, Arduino) and libraries (e.g., GPIO, PySerial) for Python-based IoT projects. Assign tasks where students write Python scripts to interface with sensors, actuators, or IoT devices for data collection and control.

7. Web Development with Python Frameworks:

- **Purpose:** Teach Python's application in building web applications.
- **Implementation:** Introduce web frameworks like Flask or Django for backend development. Assign projects where students create RESTful APIs, build CRUD (Create, Read, Update, Delete) applications, or deploy web applications using cloud platforms (e.g., Heroku, AWS).

8. Natural Language Processing (NLP) Projects:

- **Purpose:** Explore Python's capabilities in processing and analyzing textual data.
- **Implementation:** Use Python libraries such as NLTK (Natural Language Toolkit) or spaCy for tasks like text classification, sentiment analysis, or named entity recognition. Assign projects where students build NLP pipelines and develop applications for text processing.

9. Python for Scientific Computing and Simulation:

- **Purpose:** Apply Python in solving complex scientific problems and simulations.
- **Implementation:** Introduce scientific computing libraries like SciPy for numerical computations, Matplotlib for plotting, and SymPy for symbolic mathematics. Assign projects where students model physical phenomena, simulate experiments, or solve differential equations using Python.