Unit-II

Nanostructure Materials: Metals, Metal Oxides, Carbon-based nanomaterials CNT, C60, graphene. Top-down and bottom-up approach, Photolithography. Ball milling. Gas phase condensation. Vacuum deposition. Physical vapor deposition (PVD): Thermal evaporation, Chemical vapor deposition (CVD). Sol-Gel. Spray pyrolysis. Hydrothermal synthesis. Preparation through colloidal methods. MBE growth of quantum dots.

Unit-III

Characterization: X-ray diffraction, UV-Vis Spectroscopy, Photoluminescence Spectroscopy, Optical microscope vs Electron Microscopes, Scanning Electron Microscopy, Transmission Electron Microscopy. Atomic Force Microscopy. Scanning Tunnelling Microscopy.

Unit-IV

Applications: Applications of nanoparticles, quantum dots, nanowires and thin films for photonic devices (LED, solar cells), CNT-based transistors, Nanomaterial Devices: Quantum dots heterostructure lasers, optical switching and optical data storage. Magnetic quantum well; magnetic dots-magnetic data storage. Micro Electromechanical Systems (MEMS), Nano Electromechanical Systems (NEMS).

References/Books:

1. C.P. Poole, Jr. Frank J. Owens, Introduction to Nanotechnology (Wiley India Pvt. Ltd.).
2. S.K. Kulkarni, Nanotechnology: Principles & Practices (Capital Publishing Company).
3. K.K. Chattopadhyay and A. N. Banerjee, Introduction to Nanoscience and Technology (PHI Learning Private Limited).
4. Introduction to Nanoelectronics, V.V. Mitin, V.A. Kochelap and M.A. Stroscio, 2011, Cambridge University Press.
5. Richard Booker, Earl Boysen, Nanotechnology (John Wiley and Sons).

241/PHY/AE101-A

Ability Enhancement Course(s)

COURSE ID: 241/PHY/AE101

PROGRAMMING USING PYTHON

Marks (Final Exam): 35 | **Credits: 2**

Marks (Internal Assessment): 15 | **Time: 3 Hours**

Course Outcomes:

Ranjit

After successful completion of the course on Programming using Python, a student will be able to:

- *Apply various Python modules to perform numerical computations effectively.*
- *Utilize the* ***Matplotlib*** *library for visual representation and analysis of data.*
- *Implement the* ***Runge-Kutta method*** *for solving and evaluating definite integrals numerically.*
- *Conduct matrix creation, manipulation, and operations using the* ***NumPy*** *library.*

Pre-requisite Overview: Before proceeding to programming using python course, students are introduced to the foundational concepts of Python, including the Python interpreter, print statements, variables and assignments, strings, comments and documentation, debugging, input/output handling, data types and type conversion, list operations, and logical and comparison operations. Control flow mechanisms such as sequencing, iteration, selection structures (if, if-else, elif), loops (for, while), and control statements (break, continue), with practical examples like reversing a string, computing the sum of consecutive numbers, and calculating factorials. Python functions (built-in and user-defined), as well as essential data structures like lists, dictionaries, tuples, sets, and list comprehensions through real-world problem-solving exercises (e.g., hashtag generator, search engine, simple calculator). The NumPy module is then introduced for efficient array handling, covering topics such as the difference between Python lists and NumPy arrays, array creation using ones(), zeros(), random(), arange(), linspace(), along with key array operations (sum, mean, variance), matrix manipulation, reshaping, transposing arrays, and reading/writing data using savetxt() and loadtxt().

List of Programs:

1. Program to calculate mean, median and mode from a dataset.
2. Program to plot experimental data with error bars using Matplotlib.
3. Program to fit experimental data with a polynomial curve.
4. Program to find roots of a function using Newton-Raphson method.
5. Program for numerical integration using the Runge-Kutta method.
6. Program to create a 2D array (matrix) and perform basic operations like addition, subtraction, and multiplication.
7. Program for Random Walk Problem.

References/Books:

1. Python Crash Course by Eric Matthes (No Starch Press, 2nd ed., 2019).
2. Python Programming: An Introduction to Computer Science by John Zelle (Franklin, Beedle & Associates Inc., 2003).
3. Computation Physics: Problem Solving with Python, 3rd Edition by Rubin H. Landau, Manuel J Páez, Cristian C. Bordeianu (Wiley VCH, 2015).
4. Python documentation available at the Python web page (https://docs.python.org/3/).
5. Numpy documentation: https://numpy.org/doc/stable/index.html.

Ranjit