

GHAR SE CODE {

NumPy Syllabus:-

Course Title: NumPy

Course By: GSC (GHAR SE CODE)

Duration: 32 Hrs

Code Spirit: “Think in Arrays, Not Loops”

Module	Topic	Detailed Overview
1	Introduction to NumPy	- What is NumPy and why use it?- Installing NumPy- Importing the library (<code>import numpy as np</code>)- Differences between lists and NumPy arrays
2	Creating Arrays	- <code>np.array()</code> basics- Creating 1D, 2D, and 3D arrays- Using built-in functions: <code>arange()</code> , <code>linspace()</code> , <code>zeros()</code> , <code>ones()</code> , <code>empty()</code> , <code>full()</code> - Data types and <code>dtype</code>
3	Array Attributes	- <code>Shape (.shape)</code> , <code>size (.size)</code> , <code>dimensionality (.ndim)</code> , <code>itemsize</code> , and <code>datatype</code> - Reshaping arrays with <code>reshape()</code> - Flattening arrays
4	Indexing and Slicing	- Basic indexing (positive and negative)- Slicing in 1D, 2D arrays- Boolean indexing and masking- Fancy indexing with arrays of indices
5	Array Operations	- Arithmetic operations (add, subtract, multiply, divide)- Universal functions (<code>ufuncs</code>): <code>np.sqrt()</code> , <code>np.exp()</code> , <code>np.log()</code> , <code>np.sin()</code> etc.- Aggregation functions: <code>sum()</code> , <code>mean()</code> , <code>std()</code> , <code>min()</code> , <code>max()</code>
6	Broadcasting	- Concept and rules of broadcasting- Examples of broadcasting in operations- Limitations and shape compatibility

Module	Topic	Detailed Overview
7	Manipulating Arrays	- <code>reshape()</code> , <code>transpose()</code> , <code>swapaxes()</code> - Stacking arrays: <code>vstack()</code> , <code>hstack()</code> , <code>concatenate()</code> - Splitting arrays: <code>split()</code> , <code>hsplit()</code> , <code>vsplit()</code>
8	Random Module	- Introduction to <code>np.random</code> - Random number generation: <code>rand()</code> , <code>randn()</code> , <code>randint()</code> - Setting seed with <code>np.random.seed()</code> - Random sampling and permutations
9	Linear Algebra with NumPy	- Dot product and matrix multiplication (<code>dot()</code> , <code>matmul()</code>)- Identity and diagonal matrices- Determinant and inverse using <code>linalg.det()</code> and <code>linalg.inv()</code> - Solving systems of equations: <code>linalg.solve()</code>
10	Advanced Indexing and Masking	- Using conditions with <code>np.where()</code> - Filtering arrays- Conditional replacement- Complex slicing
11	Handling Missing or Invalid Data	- Use of <code>np.nan</code> , <code>np.isnan()</code> - Replacing missing values- NaN-safe functions: <code>np.nansum()</code> , <code>np.nanmean()</code>
12	Performance and Memory	- Why NumPy is faster than Python lists- Memory layout (C-style vs Fortran-style)- Time comparisons with <code>timeit</code> - In-place operations
13	File Input/Output	- Saving and loading arrays with <code>np.save()</code> , <code>np.load()</code> - Saving in text format using <code>savetxt()</code> and <code>loadtxt()</code> - Working with CSV and TSV files
14	Mini Projects / Assignments	- Basic image manipulation- Matrix transformations- Statistical simulations (dice, coin flips)- Numerical solution to equations

}