Integral University, Lucknow

SEMESTER – 1st

2. Course Name	**Robot Kinematics and Dynamics**			**L**	**T**	**P**
3. Course Code	**CS561**			3	1	0
4. Type of Course (use tick mark)				**Core (√)**	**DE ( )**	**FC ( )**
5. Pre-requisite (ifany)	none	**6. Frequency (use tickmarks)**	Even ( )	Odd (√)	Either Sem ( )	Every Sem ( )

7. Total Number of Lectures, Tutorials, Practicals

Lectures =3	**Tutorials = 1**	**Practical = 0**

8. COURSE OBJECTIVES: This course provides the basics for deriving kinematic/dynamical models of a robot (such as a factory robotic arm, mobile robot, drone, etc.) and for designing robot controllers. Students will learn how to derive the equations of motion and basic control equations themselves, and they will also learn how to use existing frameworks to do the same.

9. COURSE OUTCOMES (CO):

After the successful course completion, learners will develop following attributes:

COURSE OUTCOME (CO)	**ATTRIBUTES**
CO1	Derive the equations of motion for different types of robots
CO2	Create joint-level and endpoint-level controllers (including trajectory following)
CO3	Implement a simulation model in Python or MATLAB
CO4	Analyses the designed control systems and reflect on the design choices

10. Unit wise detailed content

Unit-1	**Number of lectures = 08**	**Title of the unit: Introduction**	**Mapped CO: 1**

Importance of the course and the big picture, Motivation with videos of examples, Connection to real world, Homogenous coordinates, quaternions. Recap of Control Theory: State-space model, root locus, bode plot, PID control, Concept of LQR and MPC control

Unit-2	**Number of lectures =08**	**Title of the unit: Endpoint Control for Robotic Manipulators**	**Mapped CO: 2**

Big picture + motivation, Difference between position and torque-controlled robots, Position control (with a PID controller), Impedance control (for position tracking tasks), Force control (for force tracking tasks)

Unit-3	**Number of lectures = 08**	**Title of the unit: Robot Arm Kinematics**	**Mapped CO: 3**

Deriving forward kinematics and Jacobian (for a 2-DoF planar arm), Transformations between endpoint and join space using Jacobian, Kinematic singularities and solution, Kinematic task priority control (for a 3-DoF planar arm), Denavit-Hartenberg convention (just an introduction), Mention parallel robots and examine differences

Unit-4	**Number of lectures = 08**	**Title of the unit: Robot Arm Dynamics**	**Mapped CO: 4**

Rigid-body dynamics model, how to derive dynamics model with Lagrange method for 2-DoF arm, Using the dynamics model in the control, Dynamic task priority control

Unit-5	**Number of lectures = 08**	**Title of the unit: Kinematics & Dynamics of Mobile Robot**	**Mapped CO: 5**

Forward kinematic models: Wheel kinematic constraints, Robot kinematic constraints, non-holonomic system, Derivation of linear bicycle model, Steady-state analysis. Dynamics & Control of Mobile Robot / Automated Vehicle: Transient analysis, Frequency response, Path-following control, Drone Robots: Dynamics of quadrotor (drone) robots, Control of quadrotor (drone) robots

11. CO-PO and PSO mapping

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	1	2	1	1	1	1	1	1	1	1	1	1	2	1	1
CO2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CO3	1	1	2	1	1	1	1	1	1	1	1	1	1	2	1
CO4	2	2	2	1	1	1	1	1	1	2	2	3	3	3	1
CO5															

3 Strong contribution, 2 Average contribution , 1 Low contribution

12. Brief description of self-learning / E-learning component

13. Books recommended:

1. Lynch, K. M., & Park, F. C. (2017). Modern robotics. Cambridge University Press.
2. Siciliano, B., Sciavicco, L., Villani, L., &Oriolo, G. (2010). Robotics: modelling, planning and control. Springer Science & Business Media.
3. Craig, J. J. (2009). Introduction to robotics: mechanics and control. Pearson Education India.
4. Ogata, K. (2010). Modern control engineering. Prentice hall.
5. Dorf, R. C., & Bishop, R. H. (2001). Modern Control Systems: Solutions Manual (Vol. 12). Prentice Hall.
6. Corke, Peter. "Robopy", Python toolbox for robotics.
7. Foote, Tully. "tf: The transform library." 2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA). IEEE, 2013.

2. Course Name	Planning and Decision Making			L	T	P
3. Course Code	CS562			3	1	0
4. Type of Course (use tick mark)				Core (✓)	DE ( )	FC ( )
5. Pre-requisite (ifany)	none	6. Frequency (use tickmarks)	Even ( )	Odd (✓)	Either Sem ( )	Every Sem ( )

7. Total Number of Lectures, Tutorials, Practicals

Lectures =3	Tutorials = 1	Practical = 0

8. COURSE OBJECTIVES: This course provides an overview of motion planning and decision-making techniques and their practical application in robotics. Planning and Decision-making are critical components of autonomy in robotic systems. These components are responsible for making decisions that range from path planning and motion planning to coverage and task planning to taking actions that help robots understand the world around them better. This course studies underlying algorithmic techniques used for planning and decision-making in robotics and examines case studies in ground and aerial robots, mobile manipulation platforms and multi-robot systems. The students will learn the algorithms and implement them in a series of programming-based projects. In particular, we will first cover the fundamentals of planning (workspace, configuration space, representation of obstacles and robot models). We will then cover a broad set of methods, which include reactive methods and feedback control; discrete, combinatorial and probabilistic planning; planning under differential constraints; planning under uncertainty; learning in planning; and planning for multi-robot systems. Finally, we will briefly touch upon task assignment and vehicle routing.

9. COURSE OUTCOMES (CO):

After the successful course completion, learners will develop following attributes:

COURSE OUTCOME (CO)	ATTRIBUTES
CO1	Explain the role of Motion Planning and Decision-Making (PDM) in Robotics, and describe possible applications
CO2	Identify different classes of robotic systems, the associated mathematical models for their kinematics, dynamics and motion and the relevant spaces for PDM, such as the configuration and state spaces
CO3	Describe and compare algorithms for motion planning and decision-making of mobile robots and multi-robot systems, including reactive algorithms, optimization-based algorithms, combinatorial algorithms, sampling-based algorithms, learning-based methods and planning in belief space
CO4	Analyse a PDM problem, consider constraints, objectives and select the appropriate PDM methods to apply
CO5	Design and implement motion planners of moderate complexity to solve motion-planning tasks and navigate mobile robots
CO6	Perform PDM experiments, evaluate the results, and draw sound conclusions. present the findings of a PDM experiment clearly in a report using graphs and scientific English. Understand PDM experiments descriptions found in scientific literature, and reproduce results

10. Unit wise detailed content

Unit-1	Number of lectures = 08	Title of the unit: Introduction	Mapped CO: 1

Planning and Decision-making in Robotics, Autonomous mobile robots, Course organization, Map & obstacles' representation

Unit-2	Number of lectures =08	Title of the unit: Fundamentals	Mapped CO: 2

3D rotations, configuration space, algorithm properties, Differential constraints: recap of modeling & control of manipulators, ground robots, aerial robots

Unit-3	Number of lectures = 08	Title of the unit: Algorithms	Mapped CO: 3

Graph-search fundamentals (depth/breadth first, Dijkstra, A*, heuristics), Combinatorial algorithms (cell decomposition, shortest-path roadmaps, motion primitives), Sampling-based algorithms (PRM, RRT, RRT*), Reactive algorithms: collision avoidance, potential fields, velocity obstacles, Kinodynamic planning

Unit-4	Number of lectures = 08	Title of the unit: Optimization based algorithms	Mapped CO: 4

Trajectory optimization I (Fundamentals + Global), Trajectory optimization II (MPC fundamentals + Local), Formal methods (reachability analysis, logic), Planning under uncertainty I (fundamentals, MDP, value iteration, Belief space), Planning under uncertainty II (POMDP, continues belief space, CCMPC), Learning in planning (deep learning, reinforcement learning, IRL), Multi-robot motion planning (joint configuration, optimization-based, coverage), Task assignment (Hungarian, auction, linear program) [1/2], Vehicle routing

11. CO-PO and PSO mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3	PSO4
CO1	1	2	3	2	1	1	2	1	2	3	2	2	3	3	1
CO2	2	3	2	2	1	1	2	1	1	2	2	1	3	3	1
CO3	2	2	3	2	1	1	2	1	1	2	3	1	3	3	1
CO4	2	1	3	3	1	1	2	1	2	2	3	2	2	3	1
CO5	2	2	2	2	1	2	1	2	2	1	1	1	2	3	1
CO6	2	1	2	1	1	1	2	1	1	1	3	1	2	2	1

3 Strong contribution, 2 Average contribution , 1 Low contribution

12. Brief description of self-learning / E-learning component

1. Planning Algorithms, S. LaValle http://planning.cs.uiuc.edu/

13. Books recommended:

2. Course Name	**Control System**			**L**	**T**	**P**
3. Course Code	**EE301**			3	1	0
4. Type of Course (use tick mark)				**Core (✓)**	**DE ( )**	**FC ( )**
5. Pre-requisite (ifany)	Linear Network & Systems	**6. Frequency (use tickmarks)**	Even ( )	Odd (✓)	Either Sem ( )	Every Sem ( )

7. Total Number of Lectures, Tutorials, Practicals

Lectures =3	**Tutorials = 1**	**Practical = 0**

8. COURSE OBJECTIVES: This course provides the concept of transfer function and mathematical modeling of systems. Students get the knowledge of first order and second order system and gain information of the system. Students will be able to evaluate the stability of the system using Nyquist stability criterion. Students will be able to design the compensator and able to analyze state space representation.

9. COURSE OUTCOMES (CO):

After the successful course completion, learners will develop following attributes:

COURSE OUTCOME (CO)	**ATTRIBUTES**
CO1	To learn the concept of transfer function and mathematical modeling of systems.
CO2	To get the knowledge of first order and second order system.
CO3	To gain information of the system.
CO4	To evaluate the stability of the system using Nyquist stability criterion
CO5	To design the compensator and also study of state space analysis.

10. Unit wise detailed content

Unit-1	**Number of lectures = 08**	**Title of the unit: Input/ Output Relationship**	**Mapped CO: 1**

Introduction to control system, Open and closed loop control system, Mathematical modeling of physical systems, Transfer function of electrical and mechanical system, Analogous systems, Block Diagram Reduction Algebra and signal flow graph, Mason's gain formula.

Unit-2	**Number of lectures =08**	**Title of the unit: Time Domain Analysis**	**Mapped CO: 2**

Time domain criteria; Test Signals; Transient and steady state response of first and second order feedback systems; Performance indices; Response analysis with proportional, Proportional- Derivative (PD) controller, Proportional-Integral (PI) controller and Proportional- Integral –Derivative (PID) controller.

Unit-3	**Number of lectures = 08**	**Title of the unit: Stability, Algebraic Criteria and Frequency response Analysis**	**Mapped CO: 3**

Asymptotic and conditional stability, Routh Hurwitz criterion, Frequency response analysis, Correlation between time and frequency domain specifications, Resonant peak, Resonant frequency, Bandwidth, Cutoff frequency, Polar plots, Bode plots.

Unit-4	**Number of lectures = 08**	**Title of the unit: Root Locus Technique and Stability in Frequency Domain**	**Mapped CO: 4**

The root locus concepts, Construction of root loci, Nyquist stability criterion, Relative stability, Gain margin, Phase margin, Constant M and N circles.

Unit-5	**Number of lectures = 08**	**Title of the unit: Introduction to Design and State variable technique**	**Mapped CO: 5**

Design through compensation Techniques; Realization of Lag, Lead, And Lag-Lead compensation; Design of closed loop control system using root locus and bode plot compensation. Introduction to State variable analysis, State space representation, State equations, State transfer matrices, Controllability and observability.

11. CO-PO and PSO mapping

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3										1	1	2	
CO2	2	2		1								3	1		
CO3	2	2										3	1	2	2
CO4	1	2	2	2								2	2		
CO5	3	2	3									3	2	2	

3 Strong contribution, 2 Average contribution , 1 Low contribution

12. Brief description of self-learning / E-learning component

1. https://archive.nptel.ac.in/courses/107/106/107106081/

13. Books recommended:

1. B. C. Kuo, "Automatic Control system", Wiley, 9th Edition, 2014.
2. I.J. Nagrath& M. Gopal, "Control system Engineering", New Age International, 4th Edition,2015.
3. K. Ogata, "Modern Control Engg.", PHI, 4th Edition, 2002.
4. S. K. Bhattacharya, "Control system Engg.", Pearson Education, 2nd Edition, 2008.
5. S. Hasan Saeed, "Automatic control system", Kataria and sons, New Delhi, 8th Edition, 2016

Course Name	Mathematical Programming			L	T	P
3. Course Code	CS546			3	1	0
4. Type of Course (use tick mark)				Core (✓)	DE ( )	FC ( )
5. Pre-requisite (ifany)	None	6. Frequency (use tickmarks)	Even ( )	Odd (✓)	Either Sem ( )	Every Sem ( )
7. Total Number of Lectures, Tutorials, Practicals						
Lectures =3		Tutorials = 1		Practical = 0		

8. COURSE OBJECTIVES:

1. Introduction to linear optimization and its extensions emphasizing the underlying mathematical structures, geometrical ideas, algorithms and solutions of practical problems.
2. how different formulations and algorithms can be combined to efficient solution methods
3. theory about linear programming, integer programming, and heuristics
4. knowledge about many different models and when they can be good starting points for modeling richer problems
5. Solving real world problems with computer software, discrete optimization formulations and algorithms

9. COURSE OUTCOMES (CO):

After the successful course completion, learners will develop following attributes:

COURSE OUTCOME (CO)	ATTRIBUTES
CO1	understand how commercial software for solving optimization problems works
CO2	understand how different ways to formulate optimization problems can affect the practical solvability of the problem
CO3	assess when optimization models might be solved by exact methods and when heuristics are needed
CO4	structure technical problems so that they can be formulated as mathematical programs
CO5	understand the pros and cons of different formulations and solution methods and the interaction between model and method

10. Unit wise detailed content

Unit-1	Number of lectures = 08	Title of the unit: Introduction	Mapped CO: 1

Mathematical Foundation: Basic Theory of Sets and Functions: Sets, Vectors, Sequences of Subsequences, Mapping and Functions, Continuous Functions; Vector Spaces; Matrices and Determinants; Linear Transformation and Rank; Convex Sets and Convex Cones, Convex and Concave Functions.

Unit-2	Number of lectures =08	Title of the unit: Linear Programming	Mapped CO: 2

Linear Programming: Definitions and Terminologies, Basic Solutions of Linear Programs, Fundamental Properties for Linear Programs; Simplex Methods: Theory of Simplex Methods, Method of Computation Replacement Operation; Degeneracy in Linear Programming: Charnes' Perturbation Method.

Unit-3	Number of lectures = 08	Title of the unit: Duality	Mapped CO: 3

Duality in Linear Programming: Canonical Dual Programs and Duality Theorems, Equivalent Dual Forms, Lagrange Multipliers and Duality, Duality in the Simplex Method; Bounded Variable Problems; Transportation Problems; Assignment Problems.

Unit-4	Number of lectures = 08	Title of the unit: Nonlinear Programming	Mapped CO: 4

Nonlinear and Dynamic Programming: Constrained and Unconstrained Optimization, Kuhn-Tucker Optimality Conditions; Quadratic Programming: Wolfe's Method, Dantzig's Method, Beale's Method, Lemke's Complementary Pivoting Algorithm.

Unit-5	Number of lectures = 08	Title of the unit: Methods of Nonlinear Programming	Mapped CO: 5

Methods of Nonlinear Programming: Separable Programming, Kelley's Cutting Plane Method, Zoutendijik's Method of Feasible Direction, Rosen's Gradient Projection Method, Zangwill's Convex Simplex Methods, Dantzig's Method for Convex Programs; Goal Programming, Multiple Objective Linear Programming, Functional Programming.

11. CO-PO and PSO mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	2	2	3	3	2	2	3	2	1	2	3	2
CO2	3	2	3	2	3	2	3	3	3	2	3	2	3	3	3
CO3	3	2	3	2	2	2	3	3	3	2	2	2	3	2	3
CO4	2	2	2	2	3	3	3	2	3	2	3	2	3	3	2
CO5	3	2	3	2	3	3	3	3	2	3	2	3	3	3	3

3 Strong contribution, 2 Average contribution , 1 Low contribution

12. Brief description of self-learning / E-learning component

13. Books recommended:

1. S. M. Sinha, Mathematical Programming: Theory and Methods, Elsevier,2005.
2. Steven Vajda, Mathematical Programming. Courier Corporation,2009.
3. Melvyn Jeter, Mathematical Programming: An Introduction to Optimization, CRCPress,1986.
4. A. Bachem, M. Grötschel, B. Korte, Mathematical Programming: The State of the Art, Springer Science & Business Media, 2012

2. Course Name	Robot Software Practical			L	T	P
3. Course Code	CS563			0	0	4
4. Type of Course (use tick mark)				Core (✓)	DE ( )	FC ( )
5. Pre-requisite (ifany)	None	6. Frequency (use tickmarks)	Even ( )	Odd (✓)	Either Sem ( )	Every Sem ( )

7. Total Number of Lectures, Tutorials, Practicals

Lectures =0	Tutorials = 0	Practical = 2

8. COURSE OBJECTIVES: This course teaches students the basics for creating and collaborating on academic robotics software. It aims to give students sufficient knowledge and experience such that they can proceed learning more about robotics programming by themselves as needed. The course does this by providing an overview of key technologies, and letting students develop basic skills often used in real world robotic software development. The main objective is to obtain hands-on experience - beginner's level - with Linux, Git&Gitlab, C++, ROS, and integrate with common external libraries such as OpenCV and PCL. This course does not teach new robotics theory (e.g., perception, planning, control, etc.) which is covered in other courses.

9. COURSE OUTCOMES (CO):

After the successful course completion, learners will develop following attributes:

COURSE OUTCOME (CO)	ATTRIBUTES
CO1	use the command line on a Linux system, perform basic file operations and job management tasks, and automate tasks with shell scripts;
CO2	use Git to commit and integrate incremental code changes together with a lab partner, and resolve any code integration conflicts; use a Gitlab server to exchange code changes, and use its issue tracker to provide and receive feedback on each other's code;
CO3	solve programming tasks in C++, compile the code using Make, debug compile-time, link-time and run-time problems;
CO4	create new C++ ROS packages, use common ROS development tools, and utilize the ROS API in their C++ programs; investigate and integrate external libraries into their own ROS components, such as OpenCV or PCL;
CO5	add code comments and basic documentation to their code, find and interpret documentation of external libraries;
CO6	develop evaluative judgments on the quality of your own work and of the performance of others (including peers) by implementing and providing feedback; collaborate and coordinate with lab partner on coding tasks.

10. Unit wise detailed content

Unit-1	Number of lectures = 08	Title of the unit: Introduction to Linux	Mapped CO: 1

Although Windows is the most prevalent operating system on desktop PCs, Linux became very popular for embedded systems such as robots. Developing for embedded Linux systems is most easily done in Linux itself, and this course aims to familiarize students with the use of Linux on the desktop and terminal. It includes the following topics: OS architecture, File system, Shell, Scripting.

Unit-2	Number of lectures =08	Title of the unit: GIT/Gitlab	Mapped CO: 2

All exercises will be submitted using Git, a widely used version control system to manage and collaborate on coding projects. During the course, students will practice basic tasks with git, such as solving code conflicts, and merging the work of lab partners. Using an issue tracker and giving constructive feedback to peers.

Unit-3	Number of lectures = 08	Title of the unit: C++	Mapped CO: 3

C++ is one of the most relevant programming languages for robotics. Being an object-oriented language brings great advantages compared to its predecessor C. This practical gives students practical knowledge on C++ programming. The course encompasses the following topics: Introduction to C++, common compilation tools, basic programming in C++, classes and objects.

Unit-4	Number of lectures = 08	Title of the unit:ROS	Mapped CO: 4

ROS: The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms. During the course, students will learn to use existing ROS tools, create their own software components in C++, integrate them into an application, and test their solutions using a physics simulator.

Unit-5	Number of lectures = 08	Title of the unit: OpenCV or PCL	Mapped CO: 5

You will also get some practice using external libraries in your robotics project, such as OpenCV or PCL. OpenCV is a computer vision library designed for computational efficiency and with a strong focus on real-time applications. The Point Cloud Library (PCL) is an open-source project for point cloud processing. Note that this course is *not* about understanding the methods or theory on which these libraries are based, but rather on integrating the provided functionality in novel software components.

11. CO-PO and PSO mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3	PSO4
CO1	1	1	2	1	1	1	2	1	2	2	1	1	1	1	
CO2	1	2	2	1	1	1	3	1	2	1	1	2	1	1	
CO3	1	2	2	1	1	1	2	1	2	2	1	2	1	2	2
CO4	1	3	3	1	2	1	3	1	2	2	1	2	2	2	
CO5	1	1	1	2	2	1	3	2	2	1	1	1	1	1	
CO6	1	1	1	2	2	2	3	1	3	2	1	1	1	1	

3 Strong contribution, 2 Average contribution , 1 Low contribution

12. Brief description of self-learning / E-learning component

1. 1A Gentle Introduction to ROS", Jason M. O'Kane. http://www.cse.sc.edu/~jokane/agitr/

13. Books recommended:

2. Course Name	Python Programming Lab			L	T	P
3. Course Code	CS272			0	0	2
4. Type of Course (use tick mark)				Core (✓)	DE ( )	FC ( )
5. Pre-requisite (ifany)	None	6. Frequency (use tickmarks)	Even ( )	Odd (✓)	Either Sem ( )	Every Sem ( )

7. Total Number of Lectures, Tutorials, Practicals

Lectures =0	Tutorials = 0	Practical = 2

8. COURSE OBJECTIVES:

1. To build a strong foundation of python and its IDEs.
2. Study of various object-oriented programming constructs and data structures available in Python.
3. Writing and using functions and modules.
4. Developing file handling applications.
5. Exploring the use of libraries in developing real-world applications.

9. COURSE OUTCOMES (CO):

After the successful course completion, learners will develop following attributes:

COURSE OUTCOME (CO)	ATTRIBUTES
CO1	Install and configure python and its IDEs.
CO2	Write basic programs using the various data structures provided in python.
CO3	Develop small modules and components using object-oriented methodology.
CO4	Use the libraries and develop file handling applications.
CO5	Develop some working applications using python.

10. Detail content

List of Experiments

1. Understanding Python installation and its Integrated Development Environments (IDEs).
2. Write a program to illustrate various data types & concepts of variables/Constant in Python.
3. Write a program to perform different Arithmetic Operations on numbers in Python (Addition, Subtraction, Multiplication, Division, etc.)
4. Write a program in python to demonstrate the concept of “Loop” and print the following pattern of prime numbers if input is number of lines. e.g.; if n=3, output should be:
5. Write a program to implement the concept of “List” (create, append, and remove lists in python).
6. Write a program to search an input number in a list of n numbers and print a “YES” along with its position (index) otherwise print a “No”.
7. Write a program to create, concatenate and print a “String” and accessing sub-string from a given string.
8. Write a program to demonstrate working of “Tuples” in python.
9. Write a program to illustrate the working of “Dictionaries” in python.
10. Write a program to check whether input string is “Pangram” or not.
11. Write a program to find factorial of a number using “Recursion”.
12. Write a program implement the concept of “Functions” in python and sort „n‟ numbers in ascending and descending order after taking input (Integer number) from user.
13. Write a program to define a “module” and import a specific function in that module to another program.
14. Write a program that reads an input text “File” and prints all of the unique words in the file in (alphabetical order).
15. Write a program that depicts the implementation of Python “Class” which reverse a string word by word.
16. Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST 2017”
17. Write a Python class to implement pow(x, n)
18. Write a program to implement the working of “NumPy” in python.

11. CO-PO and PSO mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3	PSO4
CO1	1	1	2	1	1	1	2	1	1	1	1	1	1	1	1
CO2	1	1	2	1	1	1	3	1	1	2	1	1	2	1	1
CO3	1	2	2	1	1	1	3	1	1	2	1	1	2	1	1
CO4	2	3	3	1	1	1	3	1	1	3	1	1	2	2	1
CO5	3	3	3	1	1	1	3	2	2	3	2	1	3	3	2
CO6	1	1	2	1	1	1	2	1	1	1	1	1	1	1	1

3 Strong contribution, 2 Average contribution , 1 Low contribution

12. Brief description of self-learning / E-learning component

13. Books recommended:

1. Guido van Rossum and Fred L. Drake Jr., ―An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011
2. Kenneth A. Lambert, ―Fundamentals of Python: First Programs, CENGAGE Learning,2012.
3. Timothy A. Budd, ―Exploring Python, Mc-Graw Hill Education (India) Private Ltd., 2015.
4. Robert Sedgewick, Kevin Wayne, Robert Dondero, ―Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016.