| | | |
|---|---|---|
| | **National Institute of Technology Meghalaya**<br>An Institute of National Importance | **CURRICULUM** |

| | | | |
|---|---|---|---|
| Programme | **Bachelor of Technology in Computer Science and Engineering** | Year of Regulation | **2019-20** |
| Department | **Computer Science and Engineering** | Semester | **V** |

| Course Code | Course Name | Credit Structure | | | | Marks Distribution | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | L | T | P | C | INT | MID | END | Total |
| **CS 323** | **COMPUTATIONAL GEOMETRY** | **3** | **0** | **0** | **3** | **50** | **50** | **100** | **200** |

| Course Objectives | | Course Outcomes | | |
|---|---|---|---|---|
| | To introduce techniques for designing efficient algorithms for geometric problems. | | CO1 | Develop efficient algorithms by exploiting geometric properties, and using appropriate data structures and geometric techniques. |
| | To discuss data structures used for geometric problems | | CO2 | Apply techniques and algorithms for solving problems in diversified fields like database searching, data mining, graphics and image processing, pattern recognition, computer vision, motion planning and robotics. |
| | To introduce combinatorial complexity of geometric problems. | | CO3 | Perform complexity analysis of algorithms |
| | To study rigorous algorithmic analysis of geometric problems. | | CO4 | Identify properties of geometric objects, express them as lemmas or theorems, and prove their correctness |
| | | | CO5 | Implement geometric algorithms |
| | | | CO6 | |

| No. | COs | Mapping with Program Outcomes (POs) | | | | | | | | | | | | Mapping with PSOs | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| 1 | CO1 | **3** | **3** | **0** | **1** | **0** | **0** | **0** | **0** | **2** | **0** | **0** | **0** | **3** | **0** | **3** |
| 2 | CO2 | **3** | **3** | **0** | **1** | **0** | **0** | **0** | **0** | **2** | **0** | **0** | **0** | **2** | **0** | **2** |
| 3 | CO3 | **2** | **3** | **3** | **1** | **2** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **2** | **3** | **2** |
| 4 | CO4 | **2** | **2** | **3** | **0** | **2** | **2** | **3** | **0** | **2** | **0** | **0** | **1** | **2** | **3** | **2** |
| 5 | CO5 | **2** | **2** | **3** | **0** | **2** | **2** | **3** | **0** | **2** | **0** | **0** | **1** | **3** | **3** | **3** |
| 6 | CO6 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

SYLLABUS

| No. | Content | Hours | COs |
|---|---|---|---|
| I | Geometric Preliminaries, DCEL (Doubly Connected Edge List) data structure, Polygon, Planar Straight Line Graph (PSLG) Area of a triangle, area of a polygon, Determinant used to test position of a point with respect to a directed line. Convex polygons, properties and point location in convex polygon (inside-outside test) Plane sweep algorithm, Algorithm for Line segment intersection problem using plane sweep technique. | **06** | **CO1** |
| II | Point location in PSLG – Slab method, Chain method and complexity analysis. Range Searching – 1D Range search, Kd Trees. | **06** | **CO1, CO2** |
| III | Polygon Triangulation: Regularization of polygons, properties of triangulations –Proofs, triangulation of monotone polygon – algorithm and complexity analysis. Linear Programming – Half plane intersection, Incremental algorithm and Randomized algorithm | **08** | **CO1**<br>**CO2**<br>**CO3** |
| IV | Art Gallery Theorem, Guarding Art Gallery, Fisk's proof using three colouring. Arrangements of Lines – Duality, Combinatorics of arrangements, Zone Theorem, Algorithm for Constructing arrangements of lines. | **06** | **CO3**<br>**CO4** |
| V | Convex Hulls- Convex Hull Algorithms in the Plane -Graham's Scan Algorithm, Jarvi's March, Divide and Conquer Algorithm. | **06** | **CO4**<br>**CO5** |
| VI | Voronoi Diagrams- Properties and applications in the plane. Proofs of properties related to vertices and edges of voronoi diagrams, Algorithm for constructing voronoi diagram, Delaunay Triangulation. | **08** | **CO2**<br>**CO5** |
| | Total Hours | **40** | |

**Essential Readings:**


1. Franco P. Preparata and Michael Ian Shamos, Computational Geometry an Introduction. Texts and Monographs in Computer Science, Publisher: Springer-Verlag Berlin Heidelberg, 1985, 1st Edition.
2. Joseph O'Rourke, Computational Geometry in C. Cambridge University Press, 2nd Edition, 2012.
3. Mark. de Berg, Marc. van Kreveld, Mark. Overmars and Otfried Cheong, Computational Geometry- Algorithms and Applications. Publisher: Springer-Verlag Berlin Heidelberg, 3rd Edition, 2008.


**Supplementary Readings:**


1. Herbert Edelsbrunner, Algorithms in Combinatorial Geometry, EATCS Monographs on Theoretical Computer Science, Publisher: Springer-Verlag Berlin Heidelberg, 1987, 1st Edition.
2. Joseph O' Rourke, Art Gallery Theorems, Publisher: Oxford University Press, 1987, 1st Edition.
3. De Berg, van Kreveld, Overmars, and Schwarzkopf Computational, Geometry Algorithms and Applications, Publisher: Springer-Verlag Berlin Heidelberg, 2000, 2nd Edition.