Paper Code- ES-251 **L T/P C**

Paper- Computational Lab **0 2 1**

Course Outcome:	
At the end of the course student will be able to:	
CO.ES.251.1	Utilize basic commands in Turbo C++
CO.ES.251.2	Determine the roots of the algebraic and transcendental equations using Turbo C++.
CO.ES.251.3	Develop Turbo C++ codes for implementation of interpolation formulae.
CO.ES.251.4	Develop Turbo C++ codes for solving numerical integration and differential equations

List of Experiments:

1. Program for finding roots of f(x)=0 Newton Raphson method.
2. Program for finding roots of f(x)=0 by bisection method.
3. Program for finding roots of f(x)=0 by secant method.
4. To implement Langrange's Interpolation formula.
5. To implement Newton's Divided Difference formula.
6. Program for solving numerical integration by Trapezoidal rule
7. Program for solving numerical integration by Simpson's 1/3 rule
8. To implement Numerical Integration Simpson 3/8 rule.
9. Inverse of a system of linear equations using Gauss-Jordan method.
10. Find the Eigen values using Power method.
11. Program for solving ordinary differential equation by Runge-Kutta Method.

Note: Atleast 10 experiments must be performed by the students, they may be asked to do more. Atleast 5 experiments must be from the given list.

Sr. No.	**Title of Lab Experiments**	**CO**
1.	Program for finding roots of f(x)=0 Newton Raphson method.	CO1, CO2
2.	Program for finding roots of f(x)=0 by bisection method.	CO1, CO2
3.	Program for finding roots of f(x)=0 by secant method.	CO1, CO2
4.	To implement Langrange's Interpolation formula.	CO1, CO3
5.	To implement Newton's Divided Difference formula.	CO1, CO3
6.	Program for solving numerical integration by Trapezoidal rule.	CO1, CO4
7.	Program for solving numerical integration by Simpson's 1/3 rule	CO1, CO4
8.	To implement Numerical Integration Simpson 3/8 rule.	CO1, CO4
9.	Inverse of a system of linear equations using Gauss-Jordan method.	CO1, CO4
10.	Program for solving ordinary differential equation by Runge-Kutta Method.	CO1, CO4
Extra 1	Implement an adaptive adaptive Simpson's rule, to numerically evaluate definite integrals.	
Extra 2	Develop a program to solve partial differential equations using numerical methods.	