Lab Manual
for

CSE 312 (Numerical Methods Lab)

Credit: 1, Contact hour: 2 Hours Per week

Department of Computer Science & Engineering
Varendra University
Rajshahi, Bangladesh

Varendra University
Department of Computer Science and Engineering

CSE 312
Numerical Methods Lab

Student ID	
Student Name	
Section	
Name of the Program	
Name of the Department	

Index

INSTRUCTIONS FOR LABORATORY

- The experiments are designed to illustrate about different principles, paradigm and features of numerical methods.
- Students should come with thorough preparation for the experiment to be conducted.
- Students should come with proper dress code.
- Students will not be permitted to attend the laboratory unless they bring the practical record fully completed in all respects pertaining to the experiment conducted in the previous class.
- Work quietly and carefully
- Be honest in developing and representing your program. If a particular program output appears wrong repeat the program carefully.
- All presentations of programs and outputs should be neatly and carefully done.
- If you finish early, spend the remaining time to complete the laboratory report writing.
- Come equipped with calculator and other materials related to lab works.
- Handle instruments with care. Report any breakage or faulty equipment to the Instructor. Shutdown your computer you have used for the purpose of your experiment before leaving the Laboratory.

Varendra University

Department of Computer Science and Engineering

COURSE SYLLABUS

1	**Faculty**	Faculty of Science & Engineering
2	**Department**	Department of CSE
3	**Program**	B.Sc. in Computer Science and Engineering
4	**Name of Course**	Numerical Methods Lab
5	**Course Code**	CSE 312
6	**Trimester and Year**	Summer, 2019
7	**Pre-requisites**	MAT-111
8	**Status**	Core Course
9	**Credit Hours**	1
10	**Section**	A+B
11	**Class Hours**	
12	**Class Location**	
13	**Name (s) of Academic staff / Instructor(s)**	Mst. Jannatul Ferdous, Monika Kabir
14	**Contact**	jannat@vu.edu.bd, monika@vu.edu.bd
15	**Office**	Room: 302, Eng. Annex Building, Talaimari.
16	**Counseling Hours**	
17	**Text Book**	
19	**Equipment & Aids**	1. Lab Sheet 2. Text Book 3. Calculator
20	**Course Rationale**	Numerical methods, based upon sound computational mathematics, are the basic algorithms underpinning computer

		predictions in modern systems science. Such methods include techniques for simple interpolation from the known to the unknown, linear algebra underlying systems of equations, ordinary differential equations to simulate systems. Topics covered are: the mathematical and computational foundations of the numerical approximation and solution of scientific problems; interpolation; integration and differentiation; solution of large scale systems of linear and nonlinear equations.
21	**Course Description**	This course is a study of mathematical techniques used to model engineering systems. It involves the development of mathematical models and the application of the computer to solve engineering problems using the following computational techniques: Numerical differentiation, root-finding using bracketing and open methods, linear and polynomial curve fitting, solution methods for matrix equations, numerical integration, and the solution of differential equations.
22	C**ourse Objectives**	The aim is to teach the student various topics in Numerical Analysis such as – • Solutions of nonlinear equations in one variable • Interpolation and approximation, • Numerical differentiation and integration, direct methods for solving linear systems • Numerical solution of ordinary differential equations.
23	**Learning Outcomes**	After the successful completion of this course, students will be able, I. Analyze and evaluate the accuracy of common numerical methods. II. Be able to formulate and apply numerical techniques for root finding, curve fitting, differentiation, and integration. III. Apply numerical methods to obtain approximate solutions to mathematical problems.
24	**Teaching Methods**	Lecture, Problem Solving, Brainstorming, Q/A, Project Presentation
25	**Topic Outline**	

	Class	**Topics Or Assignments**	**CLOs**	**Reading Reference**	**Activities**

	1-2	Bisection method	**I, II**	Lecture note & textbook	Problem Solving. Question Answer
	3-4	Iteration method	**I**	Lecture note and textbook	Problem Solving. Question Answer
	5-6	False-position method	**I, II**	Lecture note and textbook	Problem Solving. Question Answer
	6-8	Newton-Raphson method	**III**	Lecture note and textbook	Implementation, Problem Solving. Question Answer
	9-10	Interpolation method	**I, III**	Lecture note and textbook	Implementation, Problem Solving. Question Answer
	11-12	Lagrange method	**III**	Lecture note and textbook	Implementation, Problem Solving. Question Answer
	13-14	Matrices operations	**I**	Lecture note and textbook	Implementation, Problem Solving. Question Answer
	15-16	Simpson's rule	**III**	Lecture note and textbook	Implementation, Problem Solving. Question Answer
	17-18	Euler's method	**III**	Lecture note and textbook	Implementation, Problem Solving. Question Answer
	19-20	Final Lab Examination, Quiz, viva voce			Problem Solving, Multiple Choice Question Answer

<table>
<tr><td rowspan="9">26</td><td rowspan="9">Assessment Methods</td><td colspan="2">Assessment type</td><td>Marks</td></tr>
<tr><td colspan="2">Attendance</td><td>(10%)</td></tr>
<tr><td colspan="2">Lab Quiz</td><td>(30%)</td></tr>
<tr><td rowspan="6">Continuous assessment</td><td>Lap Report (LR)</td><td>(10%)</td></tr>
<tr><td>Regular assessment</td><td>(10%)</td></tr>
<tr><td>Lab Test (Mid)</td><td>(10%)</td></tr>
<tr><td>Lab Final (Final)</td><td>(20%)</td></tr>
<tr><td>Viva voce</td><td>(10%)</td></tr>
<tr><td>Total</td><td>(100%)</td></tr>
</table>

27	**Grading Policy**	**Grading System:**

Numerical Grade	Letter Grade	Grade Point
80 % and above	A+ (A Plus)	4.00
75% to less than 80 %	A (A Regular)	3.75
70 % to less than 75%	A- (A Minus)	3.50
65% to less than 70%	B+ (B Plus)	3.25
60% to less than 65%	B (B Regular)	3.00
55% to less than 60%	B- (B Minus)	2.75
50% to less than 55%	C+ (C Plus)	2.50
45% to less than 50%	C (C Regular)	2.25
40% to less than 45%	D (D Regular)	2.00
Less than 40%	F (Failure)	0.00

28	**Additional Course Policies**	1. *Lab Reports* Report on previous Experiment must be submitted before the beginning of new experiment. 2. *Examination* There will be lab exam in mid and at the end of the semester, that will be closed book. 3. *Unfair means policy* In case of copying/plagiarism in any of the assessments, the students involved will receive zero marks. Zero Tolerance will be shown in this regard. In case

		of severe offences, actions will be taken as per university rule. 4. *Counseling* Students are expected to follow the counseling hours posted. In case of emergency/unavoidable situations, students can e-mail me to make an appointment. 5. *Policy for Absence in Class/Exam* If a student is absent in the class for anything other than medical reasons, he/she will not receive attendance. If a student misses a class for genuine medical reasons, he/she must apply with the supporting documents (prescription/medical report). He/she will then have to follow the instructions given by the instructor for makeup. In case of absence in the mid/final exam for medical grounds, the student must also get his/her application forwarded by the head of the department before a make-up exam can be taken. It is recommended that the students inform the instructor beforehand through mail/phone call, if they feel that they will miss a class/evaluation due to medical reasons.
29	**Additional Info**	a. Academic Calendar summer 2019: *http://vu.edu.bd/notice/details/1379* b. Academic Information and Policies: *http://vu.edu.bd/program/bachelor-of-science-in-computer-science-and-engineering* c. Grading and Performance Evaluation: *http://vu.edu.bd/program/bachelor-of-science-in-computer-science-and-engineering*

Lab No.: 1

Bisection method

The bisection method is a root-finding method that applies to any continuous functions for which one knows two values with opposite sign.

This method is based on theorem, that if a function f(x) is continuous between a and b, and f(a) * f(b) < 0 i.e. f(a) and f(b) are of opposite signs, then there is exits of at least one root between a and b. Its approximate value will be given by, x_0= (a +b)/2. If f(x_0) = 0, we conclude that x_0 is a root of equation f(x) = 0. Otherwise the root lies either between x_0 and b or between a and x_0, depending on whether f(x_0) is –ve or +ve. (Considering f(a) is –ve and f(b) is +ve).

Problem:

Find the root of x^3-x-4 = 0 using Bisection method up to accuracy 0.001.

Solution:

$f(x) = x^3-x-4$

f(0) = -4, f(1) = -4. f(2) = 2.

Here, f(a) = -ve and f(b) = +ve

Root lies between interval [a, b] i.e. [1,2]

Midpoint of the interval is x0 = (a + b)/2 = (1+2)/2 = 1.5

So, f (1.5) = -2.215, this new value is giving –ve answer of f(x), so replacing the value which is already giving –ve answer of f(x) from previous interval by this new value giving –ve value of f(x) to get close interval than previous interval. Hence new interval [1.5,2]. Repeat this procedure, we get

Iteration No.	Value of a (of which f(a) is –ve)	Value of b (of which f(a) is –ve)	x_0 = (a +b)/2	f(x_0)	[a-b]
1.	1	2	1.5	-2.125	1
2.	1.5	2	1.75	-0.39	0.5
3.	1.75	2	1.875	0.716	0.25
4.	1.75	1.875	1.8125	0.141846	0.125
5.	1.75	1.8125	1.78125	-0.129608	0.0625

6.	1.78125	1.8125	1.796875	0.004803	0.03125
7.	1.78125	1.796875	1.789062	-0.062730	0.015625
8.	1.789062	1.796875	1.792969	-0.029046	0.0078125
9.	1.792069	1.796875	1.794922	-0.012142	0.0039063
10.	1.794922	1.796875	1.795898	-0.003675	0.001953

Here accuracy of 0.001953 is achieved at the end of 10^{th} iteration; hence value of root is 1.795898

Algorithm (Pseudo Code):

Decide initial values for x1 and x2, stopping criterion, E.

1. Compute f1 = f(x1) and f2 = f(x2).
2. If f1 * f2>0, x1 and x2 do not bracket any root and go to step 6;

   Otherwise continue.
3. Compute x0 = (x1+x2)/2 and compute f0 = f(x0)
4. If f1*f0 < 0 then

   set x2 = x0

   else

   set x1 = x0

   set f1 = f0
5. If absolute value of (x2 – x1)/x2 is less than error E, then

   root = (x1 + x2)/2

   write the value of root

   go to step 7

   else

   go to step 4
6. Stop

Task 1: Write a code to find lower and upper limit for bisection method.

Task 2: Find the root of given equations using this method.

Task 3: Calculate the absolute error in every step.

Task 4: Show the convergence rate graphically.

Lab No.: 2

Iteration method

An iterative method is a mathematical procedure that uses an initial guess to generate a sequence of improving approximate solutions for a class of problems, in which the n-th approximation is derived from the previous ones.

For finding the roots of the equation F(X) = 0, we rewrite this equation in the form x = Q(x). There are many ways of doing this. For ex. x3 – x2 – 1 = 0 can be expressed as,

x = (1+x)-1/2

x = (1-x3)1/2

x = (1-x2)1/3, etc.

Solution:

Given, $f(x) = x^3-x-4 = 0$

It No.	x	Q(x)
1.	1	1.7099759
2.	1.7099759	1.7873575
3.	1.7873575	1.7953954
4.	1.7953954	1.7962262
5.	1.7962262	1.796312
6.	1.796312	1.7963209

Hence root is 1.7963209

Algorithm (Pseudo Code):

```
Given an equation f(x) = 0
Convert f(x) = 0 into the form x = g(x)
Let the initial guess be x0
Do
    xi+1= g(xi)
while (none of the convergence criterion C1 or C2 is met)
```

Task 1: Find initial value for fixed point iteration.

Task 2: Find the root of given equations using this method.

Task 3: Calculate the absolute error in every steep.

Task 4: Show the convergence rate graphically.

Lab No.: 3

False position method

False position method or regula falsi is a very old method for solving an equation in one unknown, that, in modified form, is still in use. In simple terms, the method is the trial and error technique of using test ("false") values for the variable and then adjusting the test value according to the outcome.

In this method initial interval is chosen such that, f(x0) *f(x1) <0, where x_0 and x_1 are two points in which root lies.

Now eqn. pf the chord joining the two points, [x_0, f(x_0)] and [x_1, f(x_1)] is,

$$[y - f(x0)] / (x-x0) = [f(x1) - f(x0)] / (x1-x0)$$

In this method we are replacing the curve between two points [x0, f(x0)] and [x1, f(x1)] by means of the chord joining these points and taking the point of intersection of the chord with the x-axis as an approximation to the root.

Sample Problem:

Find the root of x^3-x-4 = 0, using Regula-Falsi method up to accuracy of three decimals.

Solution:

f(x) = x^3-x-4

f(0) = -4, f(1) = -4, f(2) = 2.

Root lies between interval [1,2]

x_2=1.666667

and f(x_2) = -1.037037, this is ve so replacing x_0 by x2.

So new interval is [1.6666667, 2], repeating the same procedure, we get.

It. No.	Value of x_0 (of which f(x_0) is -ve	Value of x_1 (of which f(x_1) is +ve	x_2	F(x_2)
1.	1	2	1.666667	-1.037037
2.	1.6666667	2	1.780488	-0.1360976
3.	1.7804878	2	1.794474	-0.0160251
4.	1.7944736	2	1.796107	-0.0018625
5.	1.7961073	2	1.796297	-0.0143823

Hence value of root is 1.796297

Algorithm (Pseudo Code):

1. Start
2. Read values of x0, x1 and e

   *Here x0 and x1 are the two initial guesses

   e is the degree of accuracy or the absolute error i.e. the stopping criteria*
3. Computer function values f(x0) and f(x1)
4. Check whether the product of f(x0) and f(x1) is negative or not.

   If it is positive take another initial guess.

   If it is negative then go to step 5.
5. Determine:

   x = [x0*f(x1) – x1*f(x0)] / (f(x1) – f(x0))
6. Check whether the product of f(x1) and f(x) is negative or not.

   If it is negative, then assign x0 = x;

   If it is positive, assign x1 = x;
7. Check whether the value of f(x) is greater than 0.00001 or not.

   If yes, go to step 5.

   If no, go to step 8.

   *Here the value 0.00001 is the desired degree of accuracy, and hence the stopping criteria. *
8. Display the root as x.
9. Stop

Task 1: Find initial value for false-position method.

Task 2: Find the root of given equations using this method.

Task 3: Calculate the absolute error in every steep.

Task 4: Show the convergence rate graphically.

Lab No.: 4

Newton-Raphson method

In this method the real root of the equation f(x) = 0 can be computed rapidly, when the derivative of f(x) can be easily found and is a simple expression. When an approximate value of a real root of an equation is known, a closer approximation to the root can be obtained by an iterative process, as explained below:

Let x_0 be an approximate value of a root of the equation f(x) = 0.

Let x_1 be the exact root closer to x_0, so that $x_1 = x_0 + h$, where h is small.

Since x_1 is the exact root of f(x) = 0, we have $f(x_1) = 0$, i.e., $f(x_0 + h) = 0$

For small h, by Taylor's theorem,

$$f(x0)+hf'(x0)=0$$

$$\text{and } h= - f(x0) / f'(x0)$$

$$x1=x0+h=x0-[f(x0)/f'(x0)]$$

The value of x_1 thus obtained will be a closer approximation to the actual root of f(x) = 0 than x_0. Taking x_1 as an approximate value of the root, a still better approximation x_2 can be obtained by using the formula

$$x2=x1-[f(x1)/f'(x1)]$$

The iterative process is continued until we get the required accuracy, i.e., until $|x_{n+1} - x_n|$ is less than a prescribed small value.

The iterative formula

$$x_{n+1}=x_n-[f(x_n)/f'(x_n)]$$

is called the Newton-Raphson formula.

Problem:

Find the root of x^3-x-4 = 0, using Newton Raphson method correct up to 5 decimals.

Solution:

$f(x) = x^3-x-4 = 0$,

$f'(x) = 3x^2-1$

Checking the convergence condition

f(x0)-f''(x0)/[f'(x0)]2=-0.5784<1, so, process is convergent.

f(x) = x^3-x-4, f(0) = -4, f(1) = -4, f(2) = 2.

Root lies between [1,2]

Initial guess value x_0 = (a + b) /2 = (1+2)/2 = 1.5

x1=x0-[f(x0)/f'(x0)] = 1.8695652

Repeating the same procedure, we get,

It. No.	x_0	$f(x_0)$	$f'(x_0)$	$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$
1.	1.5	-2.125	5.75	1.8695652
2.	1.8695652	0.660776	9.4858223	1.7994524
3.	1.7994524	0.027226	8.7140869	1.7963219
4.	1.796328	0.000052	8.6803825	1.7963219

Value of the root is 1.7963219

Algorithm (Pseudo Code):

1. Define f(x) = , and derivative of f(x) i.e. Df(x) =
2. Enter desired accuracy, e and initial guess, x_0
3. k = 0
4. do

   {

   x(k+1) = x(k)-[f(x(k))/Df(x(k))]

   k=k+1

   } while (|x(k+1)-x(k)|>=e)
5. Print root of the equation is, x(k+1), and no. of iterations, k
6. Stop.

Task 1: Find derivative of the given function and initial value.

Task 2: Find the root of given equations using this method.

Task 3: Calculate the absolute error in every steep.

Task 4: Show the convergence rate graphically.

Lab No.: 5

Newton's forward and backward interpolation

Interpolation is the technique of estimating the value of a function for any intermediate value of the independent variable, while the process of computing the value of the function outside the given range is called extrapolation.

Forward interpolation

If y_0, y_1, y_2, ……, y_n are the values of y = f(x) corresponding to equidistant values of x =x_0, x_1, x_2, ……, x_n, where $x_i - x_{i-1} = h$, for i = 1, 2, 3, …..,n.

Then, $y = y_0 + u\,\Delta y_0 + [u(u-1)/2!]*\Delta^2 y_0 + \ldots\ldots\ldots + [u(u-1)\ldots\ldots(u-n-1)/n!]*\Delta^n y_0$, where $u = (x - x_0)/h$.

Backward interpolation

If y_0, y_1, y_2, ……, y_n are the values of y = f(x) corresponding to equidistant values of x =x_0, x_1, x_2, ……, x_n, where $x_i - x_{i-1} = h$, for i = 1, 2, 3, …..,n, then $y = y_n + u\,\Delta y_n + [u(u+1)/2!]*\Delta^2 y_n + \ldots\ldots\ldots + [\{u(u+1)\ldots\ldots(u+n-1)\}/n!]*\Delta^n y_n$, where $u = (x - x_n)/h$.

Sample Problem:

Using the following table find out the values for y at x = 3.5, 7.7 by Newton’s forward & backward interpolation formula.

x	0	1	2	3	4	5	6	7
y	1	2	4	7	11	16	22	29

Task 1: Find the difference table from the given value of x and y.

Task 2: Write a program find the value of equal space from the data.

Task 3: Write a program to find the missing value of y for any specific value of x using Newton’s Forward Interpolation.

Task 4: Write a program to find the missing value of y for any specific value of x using Newton’s Backward Interpolation.

Lab No.: 6

Gauss's central interpolation

The gaussian interpolation comes under the Central Difference Interpolation Formulae which differs from Newton's Forward interpolation formula formula. Then the process of finding the value of y corresponding to any value of x=xi between x0 and xn is called interpolation.

If $y_0, y_1, y_2, \ldots\ldots, y_n$ are the values of $y = f(x)$ corresponding to equidistant values of $x = x_0, x_1, x_2, \ldots\ldots, x_n$, , for $i = 1, 2, 3, \ldots\ldots, n$, then

$$y = y_0 + G_1\,\Delta y_0 + G_2\,\Delta^2 y_{-1} + \ldots\ldots$$

and $G_1=p$, $G_2=p(p-1)/2$, $G3=(p+1)p(p-1)/3!$ and so on.

Sample Problem:

Using the following table find out the values for y at x = 3.5,4.5 by Gauss's forward & backward interpolation formula.

x	0	1	2	3	4	5	6	7
y	1	2	4	7	11	16	22	29

Task 1: Find the difference table from the given value of x and y.

Task 2: Write a program find the value of equal space from the data.

Task 3: Write a program to find the missing value of y for any specific value of x using Gauss's Central Forward Interpolation.

Lab No.: 7

Lagrange's interpolation

In numerical analysis, Lagrange polynomials are used for polynomial interpolation. For a given set of points (xj,yj) with no two xj values equal, the Lagrange polynomial is the polynomial of lowest degree that assumes at each value xj the corresponding value yj (i.e. the functions coincide at each point).

Algorithm (Pseudo Code)

```
1. Set sum=0
2. For i=0 to n step 1 do
3. Set p=1
4. For j=0 to  n step 1 do
 If i<> j then
 Set p=p*(x-xj)/(xi-xj)
 End if
 End do j
5. Set sum = sum + p · f
 End do i
6. Print interpolated value is sum
 End
```

Sample Problem:

Using the following table find out the values for y at x = 2 by lagrange's interpolation formula.

x	0	1	2	3	4
y	-12	0	--	12	24

Task 1: Find the value of y for corresponding x using Lagrange's interpolation formula.

Task 2: Find the value of x for corresponding y using Lagrange's inverse interpolation formula.

Lab No.: 8

Matrix addition & multiplication

Algorithm (Pseudo Code)

1. Start
2. Declare variables and initialize necessary variables
3. Enter the element of matrices by row wise using loops
4. Check the number of rows and column of first and second matrices
5. If number of rows of first matrix is equal to the number of columns of second matrix, go to step 6. Otherwise, print matrix multiplication is not possible and go to step 3.
6. Multiply the matrices using nested loops.
7. Print the product in matrix form as console output.
8. Stop

Task 1: Take inputs as matrix using array and print them**.**

Task 2: Add two matrixes and print their result.

Task 3: Multiply two matrixes and print the result.

Lab No.:9

Matrix inversion

For a given matrix equation like AX = C, where A and C are given and are told to figure out X, you would like to "divide off" the matrix A. On the other hand, to find the inverse of A, something similar to finding the reciprocal fraction above. The inverse of A, written as "A–1" and pronounced "A inverse", would allow you to cancel off the A from the matrix equation and then solve for X.

Steps:

AX = C
A–1AX = A–1C
IX = A–1C
X = A–1C

Task 1: Find the transpose of a given matrix.

Task 2: Find adjugate (adjoint) of a given matrix.

Task 3: Find the inversion of the given matrix.

Lab No.: 10

Simpson's rule

In numerical analysis, Simpson's rule is a method for numerical integration, the numerical approximation of definite integrals. Specifically, it is the approximation for equally spaced subdivisions.

Rules:

Simpson's 1/3 rule:

$A=h/3\ [y_0+4(y_1+y_3+y_5+\ldots.)+2(y_2+y_4+y_6+\ldots.)]$

Simpson's 3/8 rule:

$A=3h/8\ [y_0+(3y_1+3y_2+y_3)+(y_3+3y_4+3y_5+y_6)+\ldots\ldots.]$

Task 1: Find area of given curve using Simpson's rule.

Lab No.: 11

Euler's method

Euler's method is a numerical method to solve first order first degree differential equation with a given initial value. It is the most basic explicit method for numerical integration of ordinary differential equations. The Euler method is a first-order method, which means that the local error (error per step) is proportional to the square of the step size, and the global error (error at a given time) is proportional to the step size. The Euler method often serves as the basis to construct more complex methods.

Algorithm (Pseudo Code)

1. define f(t, y)
2. input t0 and y0.
3. input step size, h and the number of steps, n.
4. for j from 1 to n do
   i. m=f(t0,y0)
   ii. y1=y0+h∗m
   iii. t1=t0+h
   iv. Print t1 and y1
   v. t0=t1
   vi. y0=y1
5. end

Task 1: Solve a first order differential equation using Euler's method.

Lab 12:

Lab Final Test

Task 1: Quiz
Task 2: Solving the given problem.
Task 3: Viva Voce.
Task 4: Lab report evaluation

<table>
<tr><th colspan="3">Assessment and Marks Distribution</th></tr>
<tr><th colspan="2">Assessment type</th><th>Marks</th></tr>
<tr><td colspan="2">Attendance</td><td>(10%)</td></tr>
<tr><td colspan="2">Lab Quiz</td><td>(30%)</td></tr>
<tr><td rowspan="6">Continuous assessment</td><td>Lap Report (LR)</td><td>(10%)</td></tr>
<tr><td>Regular assessment</td><td>(10%)</td></tr>
<tr><td>Lab Test (LT)</td><td>(10%)</td></tr>
<tr><td>Lab Final (LF)</td><td>(20%)</td></tr>
<tr><td>Viva voce</td><td>(10%)</td></tr>
<tr><td>Total</td><td>(100%)</td></tr>
</table>

Mst. Jannatul Ferdous
Lecturer, Dept. of CSE
Varendra University, Rajshahi.

Monika Kabir
Lecturer, Dept. of CSE
Varendra University, Rajshahi.