Laboratory Manual

(Version 9.0)

for

C# Programming-Lab

(MCA-255)

MCA - III Semester

Compiled by:

Mr. Uttam Singh Bist

(Assistant Professor, BVICAM, New Delhi)

Bharati Vidyapeeth's
Institute of Computer Applications and Management (BVICAM)

A-4, Paschim Vihar, Rohtak Road, New Delhi-63

Visit us at: www.bvicam.in

Index

List of Abbreviations

BTL	Bloom's Taxonomy Level
CE	Communication Efficacy
CICP	Conduct Investigations of Complex Computing Problems
CK	Computational Knowledge
CO	Course Outcome
DAC	Departmental Advisory Committee
DDS	Design and Development of Solutions
I&E	Innovation and Entrepreneurship
I&T	Individual & Team Work
IQAC	Internal Quality Assurance Cell
LLL	Life-Long Learning
MTU	Modern Tool Usage
PA	Problem Analysis
PE	Professional Ethics
PEO	Programme Educational Objective
PMF	Project Management and Finance
PO	Programme Outcome
SEC	Societal and Envoirnmental Concern

Declaration

Department	:	Department of Computer Science and Applications
Course, Year and Semester to which Lab if offered	:	MCA II Year, III Semester
Name of Course/Lab	:	C# Programming Lab
Course Code	:	MCA-255
Version No.	:	9.0
Name of Course/Lab Teacher with Designation	:	Mr. Uttam Singh Bist, Assistant Professor
Laboratory Manual Committee	:	1. Mrs. Vaishali Joshi, Chairperson 2. Dr. Anupam Baliyan, Member 3. Dr. Ritika Wason, Member 4. Mrs. Tanya Pathak Garg, Member 5. Mr. Uttam Singh Bist, Member 6. Prof. P. S. Grover, Margdarshak 7. Mr. Amit Sharma, Alumni & Industry Expert 8. Dr. Ritika Wason, Concerned Subject Teacher, Convener
Approved by	:	DAC Date: ??/??/2019
Approved by	:	IQAC Date: ??/??/2019

Signature (Course Teacher)	**Signature** (Head of Department)	**Signature** (IQAC Coordinator)

1. Vision of the Department

To become a Centre of excellence in the field of Computer Science and Applications, to contribute effectively in the rapidly changing global economy directed towards national development ensuring prosperity for the mankind.

2. Mission of the Department

M_1 To become a centre of excellence in the field of Computer Science and Applications and produce professionals as per global industry standards.

M_2 To foster innovation, entrepreneurial skills, research capabilities and bring all-round development amongst budding professionals.

M_3 To promote analytical and collaborative life-long learning skills, among students and faculty members involving all stakeholders.

M_4 To inculcate strong ethical values and professional behaviour while giving equal emphasis to social commitment and nation building.

3. Programme Educational Objectives (PEOs)

The PEO's for the MCA programme are as follows:

PEO_1 Exhibit professional competencies and knowledge for being a successful technocrat.

PEO_2 Adopt creative and innovative practices to solve real-life complex problems.

PEO_3 Be a lifelong learner and contribute effectively to the betterment of the society.

PEO_4 Be effective and inspiring leader for fellow professionals and face the challenges of the rapidly changing multi-dimensional, contemporary world.

4. Programme Objectives (POs)

PO_1: Computational Knowledge (CK)

Demonstrate competencies in fundamentals of computing, computing specialisation, mathematics, and domain knowledge suitable for the computing specialisation to the abstraction and conceptualisation of computing models from defined problems and requirements.

PO_2: Problem Analysis (PA)

Identify, formulate, and analyze complex real-life problems in order to arrive at computationally viable conclusions using fundamentals of mathematics, computer sciences, management and relevant domain disciplines.

PO_3: Design and Development of Solutions (DDS)

Design efficient solutions for complex, real-world problems to design systems, components or processes that meet the specifications with suitable consideration to public health, and safety, cultural, societal, and environmental considerations.

PO_4: Conduct Investigations of Complex Computing Problems (CICP)

Ability to research, analyze and investigate complex computing problems through design of experiments, analysis and interpretation of data, and synthesis of the information to arrive at valid conclusions.

PO_5: Modern Tool Usage (MTU)

Create, select, adapt and apply appropriate technologies and tools to a wide range of computational activities while understanding their limitations.

PO_6: Professional Ethics (PE)

Ability to perform professional practices in an ethical way, keeping in mind cyber regulations & laws, responsibilities, and norms of professional computing practices.

PO7: Life-Long Learning (LLL)

Ability to engage in independent learning for continuous self-development as a computing professional.

PO8: Project Management and Finance (PMF)

Ability to apply knowledge and understanding of the computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects in multidisciplinary environments.

PO9: Communication Efficacy (CE)

Ability to effectively communicate with the technical community, and with society at large, about complex computing activities by being able to understand and write effective reports, design documentation, make effective presentations, with the capability of giving and taking clear instructions.

PO10: Societal and Envoirnmental Concern (SEC)

Ability to recognize and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities applicable to professional computing practices.

PO11: Individual & Team Work (I&T)

Ability to work in multi-disciplinary team collaboration both as a member and leader as per need.

PO12: Innovation and Entrepreneurship (I&E)

Ability to apply innovation to track a suitable opportunity to create value and wealth for the betterment of the individual and society at large.

5. Institutional Policy for Students' Conduct

The following guidelines shall be followed:-

5.1 All the students in their introductory Lab. shall be assigned a system, which shall be their workplace for the complete semester. Students can store records of all their Lab. assignments on their individual workstations.

5.2 Introductory Lab. shall include an introduction to the appropriate software/tool, followed by a basic Introductory Assignment having Practice Questions. All the students are expected to complete this assignment within a week time, as the same shall be assessed through a lab. test.

5.3 Each week the instructor, in parallel to respective topics covered in the theory lecture, shall assign a set of practical problems to the students in form of Assignments (A, B, C, ...). The problems in these assignments shall be divided into two parts. The first set of Problems shall be compulsory for all the students and its record need to be maintained in the Prcatical File, having prescribed format, as given in Appendix-A. All the students should get the weekly assigntment checked and signed in the Practical File by the respective teacher in the immediate succeeding week. The second set of problems are Advanced Problems and shall be optional. Student may solve these advanced problems for their further practice.

5.4 Cellular phones, pagers, CD players, radios and similar devices are prohibited in the classrooms, laboratories and examination halls.

5.5 Laptop-size computers / Tablets may be used in lectures for the purpose of taking notes or working on team-projects.

5.6 The internal practical exam shall be conducted towards the end of the semester and shall include the complete set of Lab exercises conducted as syllabus. However, students shall be assessed on continuos basis through overall performances in regular lab. tests, both announced and surprise and viva-voce.

5.7 The respective faculty shall prepare and submit sufficient number of practical sets of computing problems to the Dean (Examinations), atleast two weeks prior to the actual exam. It is the responsibility of the faculty to ensure that a set should not be repeated for more than 5 students in a given batch.

5.8 The exam shall be of 3 hours duration where the student shall be expected to implement solutions to his/her assigned set of problems on appropriate software tools in the lab.

5.9 Once implemented, student shall also appropriately document code implemented in the assigned answer sheets, which shall be submitted at the end of the examination. All the students shall also appear for viva-voce examination during the exam.

5.10 Co-operate, Collaborate and Explore for the best individual learning outcomes but copying or entering into the act of plagiarism is strictly prohibited.

6. Learning Outcomes of Laboratory Work

The student shall demonstrate the ability to:

- ☑ Verify and Implement the concepts and theory learnt in class.
- ☑ Code and use Software Tools to solve problems and present their optimal solutions.
- ☑ Apply numerical/statistical formulas for solving problems/questions.
- ☑ Develop and apply critical thinking skills.
- ☑ Design and present Lab as well as project reports.
- ☑ Apply appropriate methods for the analysis of raw data.
- ☑ Perform logical troubleshooting as and when required.

- ☑ Work effectively as a member of a team in varying roles as need be.
- ☑ Communicate effectively, both oral and written.
- ☑ Cultivate ethics, social empathy, creativity and entrepreneurial mindset.

7. Course/Lab Outcomes (COs)

CO_1 Apply logical thinking to develop programs in C for different algorithms using basic building blocks of the C language.

CO_2 Develop efficient programs using advanced data types, pointers and dynamic memory allocation functions.

CO_3 Implement real-world computing solutions through appropriate usage of the pre-processor as well as file handling on Ubuntu environment, using library functions and system calls.

CO_4 Apply C constructs to programming problems to control, manipulate files, directories and processes on Ubuntu.

CO_5 Work in teams to develop project for real-life cases.

8. Mapping of CO's with PO's

Table 1: Mapping of CO's with PO's

PO/CO	PO_1	PO_2	PO_3	PO_4	PO_5	PO_6	PO_7	PO_8	PO_9	PO_{10}	PO_{11}	PO_{12}
CO_1	3	2	-	-	3	3	-	-	-	3	-	-
CO_2	3	3	3	-	-	3	-	-	-	3	-	-
CO_3	3	3	3	-	2	3	-	-	-	3	-	-
CO_4	3	3	3	-	-	3	2	-	-	3	-	-
CO_5	3	3	3	2	3	3	2	3	3	3	3	2

9. Course/Lab Description

Course (Lab) Title : C# Programming-Lab

Course (Lab) Code : MCA-255

Credits : 02

Pre-requisites : OOPs Concepts,C++ Programming

Academic Session : July to December

Contact Hours/Week : 02 (01 Labs of 02 hours each/Week)

Internal Assessment : 40 Marks

External Assessment : 60 Marks

10. Grading Policy

Item	Points	Marks	Remarks
Weekly Lab Assignments including Practical Files	10	10	Closed Book/Open Book
Internal End-Term Practical Examination	20	10	Closed Book
Viva-Voce	10	10	Closed Book
Project	10	10	Innovative Applications of Programming
External End-Term Examinations	60	60	Closed Book (conducted and evaluated by the University)
	Total	**100**	

11. Lesson Plan

Week No.	Lab No.	Topics / Concepts to be Covered	Reference of the Lab Manual
1.	1.	C# introduction and IL Code	Assignment A (Problem AP1 to AP4)

Week No.	Lab No.	Topics / Concepts to be Covered	Reference of the Lab Manual
2.	2.	Basic OOPs programming and implementation in C# using Array	Assignment B (Problem BP1 to BP6)
3.	3.	Exploring the features of static keyword	Assignment C (Problem CP1 to CP4)
4.	4.	Create and Implement the solution in C# with indexers	Assignment D (Problem DP1 to DP2)
5.	5.	Revision	Assignment A-D
6.	6.	Identify and Implement the Exceptions in any given problems.	Assignment E (Problem EP1 to EP4)
7.	7.	Understand the use of properties to define the constraints	Assignment F (Problem FP1)
8.	8.	Interfaces	Assignment G (Problem GP1)
9.	9.	Revision	Assignment E-G
10.	10.	Delegates	Assignment H (Problem HP1 to HP5)
11.	11.	Genericss	Assignment I (Problem IP1 to IP2)
12.	12.	LINQ	Assignment J (Problem JP1 to JP5)
13.	13.	Buffer Reserved for Revision	Assignment H-J

12. Assignments

***Assignment Set*: A**

Objectives of the Assignment:

- Familiarize with the concepts of OOPs in C#-programming.
- Familiarize with the construction of IL Code Generation and compile multiple IL into single Library.
- Test and execute the basic class construction programs and correct syntax and logical errors.

CO/BTLCovered: CO_1/BTL_2 & BTL_3

Problems:

AP_1 Define a class Student, which contains the following information about students: full name, course, subject, university, e-mail and phone number.

AP_2 Declare several constructors for the class Student, which have different lists of parameters (for complete information about a student or part of it). Data, which has no initial value to be initialized with null. Use nullable types for all non-mandatory data.

AP_3 Add a method in the class Student, which displays complete information about the student.

AP_4 Generate the IL code for the above mentioned class independently. Compile the multiple classes' IL into single executable file.

Advanced Problems:

AA_1 A company pays its employees on a weekly basis. The employees are of four types:

1. Salaried employees are paid a fixed weekly salary regardless of the number of hours worked
2. Hourly employees are paid by the hour and receive overtime pay for all hours worked in excess of 40 hours
3. Commission employees are paid a percentage of their sales
4. Salaried-Commission employees receive a base salary plus a percentage

of their sales.

For the current pay period, the company has decided to reward salaried-commission employees by adding 10% to their base salaries. The company wants to implement a C# application that performs its payroll calculations polymorphic way.

a. Design the class Diagram.
b. Implement the code to fulfil the requirement.
c. Calculation must be done with polymorphic way.

Assignment Set: B Hidden Class Concepts and Logical Implementation

Objectives of the Assignment:

- Familiarize with problem solving using basic OOPs Concepts in C# Programming.
- Familiarize with the concepts of Array in C# Programming.
- Familiarize with the concepts to reconstructs simple modular programs into OOPs based solutions.

CO/BTLCovered: CO_1/BTL_2 & BTL_3

Problems:

BP$_1$ Define a class, which contains information about a mobile phone: model, manufacturer, price, owner, features of the battery (model, idle time and hours talk) and features of the screen (size and colors).

BP$_2$ Declare several constructors for each of the classes created by the previous task, which have different lists of parameters (for complete information about a student or part of it). Data fields that are unknown have to be initialized respectively with null or 0

BP$_3$ To the class of mobile phone in the previous two tasks, add a static field nokiaN95, which stores information about mobile phone model Nokia N95. Add a method to the same class, which displays information about this static field.

BP$_4$ Add an enumeration BatteryType, which contains the values for type of the battery (Li-Ion, NiMH, NiCd, …) and use it as a new field for the class Battery.

BP$_5$ Add a method to the class GSM, which returns information about the object as a

string.

BP_6 Define properties to encapsulate the data in classes GSM, Battery and Display.

Advance Problems:

BA_1 Recent events in the stock market may seem remote to you, but they underscore the uncertainty of planning for the future. People who had been thinking of retiring in the next year or so may have to rethink those plans, as the value of their 401K accounts drops noticeably. Although retirement may seem a long way off for you, we are going to explore some simple ideas in accruing funds. We are going to start with a simple model of saving for retirement. Many Departments will contribute the equivalent of 5% of your salary to a retirement fund, and then will match any contribution that you add, Rupees for Rupees, up to an additional 5%. Thus, you can salt away up to 15% of your salary into a retirement account (10% which comes as a bonus from your employer). We can model a retirement fund with some simple equations. Assume your starting salary is represented by salary; that the percentage of your salary you put into a retirement fund is ; and that the annual growth percentage of the retirement fund is growthRate i.e. 2% initial. Then your retirement fund, represented by the list F, should increase as follows:

Retirement Funds	
End of year 1	*F[0] = salary * save * 0.01*
End of year 2	*F[1] = F[0] * (1 + 0.01 * growthRate) + salary * save * 0.01*
End of year 3	*F[2] = F[1] * (1 + 0.01 * growthRate) + salary * save * 0.01*

Problem 1. This project may be used by Different Department of the Govt. Officials. So, we need to implement the Code in C# to evaluate the Retirement funds of the employees at the time of the retirement. Maximum retirement age of the Employee should be 60-65 Years as per Department requirement. Design and implement all the classes with full functionality to evaluate the retirement funds.

Assignment Set: C

Objectives of the Assignment:

- Familiarize with problem solving using basic constructs of C# Programming.

- Implement the given problem, analyse the problem and implement an appropriate solution for the problem.
- Decompose a problem into functions and synthesize a complete program using divide and conquer approach.

CO/BTLCovered: CO_1 & CO_2/BTL_2 & BTL_3

Problems:

CP_1 Create a class Call, which contains information about a call made via mobile phone. It should contain information about date, time of start and duration of the call.

CP_2 Add a property for keeping a call history – CallHistory, which holds a list of call records.

CP_3 In GSM class add methods for adding and deleting calls (Call) in the archive of mobile phone calls. Add method, which deletes all calls from the archive.

CP_4 In GSM class, add a method that calculates the total amount of calls (Call) from the archive of phone calls (CallHistory), as the price of a phone call is passed as a parameter to the method.

Advanced Questions:

CA_1 BPO Problem A company ABC Pvt. Ltd. is a Customer care BPO company, which provide its services to some Laptop leading organization (e.g. HP, COMPAQ, ASUS, DELL etc.). Now ABC Pvt. Ltd. declares that the company has 100 customer care executives, attend the 100 distinct customers' calls at a time. Now, ABC Pvt. Ltd. need a software which state that how many callers are currently involving in to attending the calls and how many lines are still free so that they can route the call towards the free lines. The Company also desires that the software automatically calculate the bills to its Organizational Customers (i.e. HP, ASUS, COMPAQ, DELL etc.).

Problem 1. As we already know that the company has 100 Customer Executive in same shift. Every time when a customer makes a call to the ABC Pvt. Ltd. then one line goes to engage and 99 left open for the next customer. At the time of hang up the phone the line become free and available to attend the next call.

Problem 2. While a customer makes a call to the customer care then the executive has to ask the Laptop Company Name. So that the System may calculate the cost of call to attend the customer in respect of Client Company. ABC is not going to charge any call cost from the Clients Company's Customer. ABC charges Rs. 1 for every 3 minutes, >3 minutes charges become Rs. 2.50, >5 minutes charges will be Rs. 4.50.

Assignment Set: D

Objectives of the Assignment:

- Familiarize with problem solving using basic constructs of C# Programming.
- Create and Implement the solution in C# with indexers.

CO/BTLCovered: CO_1 & CO_2/BTL_2 & BTL_3

Problems:

DP$_1$ Create a class Session, which can store the details of user into session while user get logged into the application. The session must have implement an indexer to fulfil the requirement.

DP$_2$ User can retrieve the details by providing the index name into the session object.

Advanced Problems:

DP$_1$ The stock (also capital stock) of a corporation constitutes the equity stake of its owners. It represents the residual assets of the company that would be due to stockholders after discharge of all senior claims such as secured and unsecured debt.

Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit.

The Stock Market prediction task is interesting as well as divides researchers and academics into two groups those who believe that we can devise mechanisms to predict the market and those who believe that the market is efficient and whenever new information comes up the market absorbs it by correcting itself, thus there is no space for prediction.

Problem 1: How to Predict Stock Value? You need to identify all the classes and

explicitly invoke the method for desired prediction.

Method Momentum

1. Divide today's close by the close a certain number of days ago.

For example, you can look back five days.

2. Multiply that number by 100.

M = (Price Today/Price Five Days Ago) x100

M = (15/10) x 100 = 150

Method Mean Reversion (Basic)

Find out the mean of last X days then evaluate the correlation between today's price and mean price to predict the values of stock for the next day.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$\sigma_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad \sigma_y = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$\sigma_{x,y} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$$\rho_{x,y} = \frac{\sigma_{x,y}}{\sigma_x \sigma_y}$$

[**Hint**: Indexer must be used in this problem statement]

Assignment Set: E

Objectives of the Assignment:

- Familiarize with problem solving in C# Programming and understanding the concept of exception handling.
- Identify the Exceptions and Implement the Exception handling mechanism in any given problem.

CO/BTLCovered: CO_1 & CO_2/BTL_2 & BTL_3

Problems:

EP1 Implement a program that takes a positive integer from the console and prints the square root of this integer. If the input is negative or invalid print "Invalid Number" in the console. In all cases print "Good Bye".

EP$_2$ Write a method ReadNumber(int start, int end) that reads an integer from the console in the range [start…end]. In case the input integer is not valid or it is not in the required range throw appropriate exception. Using this method, write a program that takes 10 integers a1, a2, …, a10 such that $1 < a_1 < \ldots < a_{10} < 100$.

EP$_3$ Implement a program that takes as a parameter the name of a text file, reads the file and returns its content as string. What should the method do if and exception is thrown?

EP$_4$ Implement a program that takes as a parameter the name of a binary file, reads the content of the file and returns it as an array of bytes. Write a method that writes the file content to another file. Compare both files.

Advanced Problems:

EA1 Exceptions are known as run time errors. We need to handle these exceptions while writing a program, because these exceptions lead to crash your program. A good software engineer must know how to manage the exceptions raising at the time of implementation. In the previous Problem statement, we assume that we have already developed and deployed those applications to our stakeholders. The stakeholders reported us that the programs or application modules are misbehaving for different inputs. Our Stakeholders want to find a sound result to improve the productivity of the application and minimizing the mistakes during the implementation.

The problem has been transferred to our analyst team. The team has shortlisted some sort of Exception that may precede a cause to crash the program. Now you need to identify and implement those Exception from the given list, which exceptions need to handle for the smooth function of the application.

The Exception Identification is required for all the previous Problems i.e. Problem No. BA1, CA1, DA1.

Exception List Proposed by the Analyst Team.

1. StackOverflow Exception
2. NullReference Exception
3. InvalidCastType Exception
4. ArrayTypeMismatch Exception

5. IndexOutOfRange Exception
6. AccessVoilationException
7. InvalidOperation Exception
8. ArgumentException
9. ArgumentNullException
10. ArgumentOutOfRangeException
11. OverflowException
12. OutOfMemoryException

Assignment Set: F

Objectives of the Assignment:

- Familiarize with problem solving in C# Programming and understanding the use of properties to define the constraints in any given problem to avoid system failure.
- Create and Implement the solution in C# with appropriate properties in Classes.

CO/BTLCovered: CO_1 to CO_3/BTL_2 to BTL_6

Problems:

FP$_1$ A company pays its employees on a weekly basis. The employees are of four types:

1. Salaried employees are paid a fixed weekly salary regardless of the number of hours worked,
2. Hourly employees are paid by the hour and receive overtime pay for all hours worked in excess of 40 hours
3. Commission employees are paid a percentage of their sales
4. Salaried-Commission employees receive a base salary plus a percentage of their sales.

For the current pay period, the company has decided to reward salariedcommission employees by adding 10% to their base salaries. The company wants to implement a C# application that performs its payroll

calculations polymorphically.

a. Identify and declare all the properties of an employee with suitable constraints, e.g. Age could not be less than 18 years and not exceeds than 60 years.
b. Salary of a salaried employee won't be exceeded Rs. 20000 per week and not less than Rs.4000 per week. The joining of an employee would not be less than Rs.4000 per week. Every year salaried employees get an increment of 10%.
c. Hourly Employee cannot work less than 30 hrs. in a week and not more than 50 hrs. in a week. Minimum and maximum payment is Rs. 200 and 400 per hour respectively.
d. The commissioned Employees' commission cannot exceed Rs. 20000 in a week. The commission on per article sale is 10%, which is fixed.
e. No one employee can earn more than Rs. 25000 per week.

Advanced Problems:

FA1 We have a school. In school we have classes and students. Each class has a number of teachers. Each teacher has a variety of disciplines taught. Students have a name and a unique number in the class. Classes have a unique text identifier. Disciplines have a name, number of lessons and number of exercises. The task is to shape a school with C# classes. You have to define classes with their fields, properties, methods and constructors. Also define a test class, which demonstrates, that the other classes work correctly.

Assignment Set: G

Objectives of the Assignment:

- Familiarize with problem solving in C# Programming and understanding the use of interfaces.
- Create and Implement the solution in C# with appropriate implementation of interfaces in Class.

CO/BTLCovered: CO_1 to CO_3/BTL_2 to BTL_6

Problems:

GP1 The stock (also capital stock) of a corporation constitutes the equity stake of its owners. It represents the residual assets of the company that would be due to stockholders after discharge of all senior claims such as secured and unsecured debt.

Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit.

The Stock Market prediction task is interesting as well as divides researchers and academics into two groups those who believe that we can devise mechanisms to predict the market and those who believe that the market is efficient and whenever new information comes up the market absorbs it by correcting itself, thus there is no space for prediction.

Problem 1: How to Predict Stock Value? You need to identify all the classes and explicitly invoke the interfaces for desired prediction.

Interface Momentum

1. Divide today's close by the close a certain number of days ago.

For example, you can look back five days.

2. Multiply that number by 100.

M = (Price Today/Price Five Days Ago) x100

M = (15/10) x 100 = 150

Interface Mean Reversion (Basic)

Find out the mean of last X days then evaluate the correlation between today's price and mean price to predict the values of stock for the next day.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$\sigma_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad \sigma_y = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$\sigma_{x,y} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$$\rho_{x,y} = \frac{\sigma_{x,y}}{\sigma_x \sigma_y}$$

[**Hint**: Indexer must be used in this problem statement]

Assignment Set: H

Objectives of the Assignment:

- Familiarize with delegates in C# Programming and understanding the use of delegates in application.
- Create and Implement the delegates in C#.

CO/BTLCovered: CO_1 to CO_3/BTL_2 to BTL_6

Problems:

HP_1 Implement the Code in C# to Display the index of a searched item from an array using Delegates.

HP_2 Implement the Code in C# to display the elements of an array using Delegates.

HP_3 Implement the Code in C# to demonstrate the to Combine Two Delegates method in GP 1 and GP 2.

HP_4 Implement the Code in C# to sort and display the elements of an array (use Bubble Sort Algorithm) using Anonymous Method.

HP_5 Implement the Code in C# to search an Item from an array using Lambda Expression. The array and Lambda are in distinct class. Lambda Expression return the value either true/false.

Assignment Set: I

Objectives of the Assignment:

- Familiarize with Generics in C# Programming and understanding the reusability of generics in an application.

CO/BTLCovered: CO_1 to CO_3/BTL_2 to BTL_6

Problems:

Classes to be implemented in C# and perform the following actions in classes.

Employee { ID, Name, Address, salary}

Student {Rollno, Name, Course, Fees}

IP$_1$ Write a generic class of list named MYLIST<T> to create the list of only either employee or students. [Note: Don't use List<T> which is system defined, But you can inherits the List<T> into MyList<T> class.] .

IP$_2$ Implement the following functionality into the custom list program.

a. Add(T): to add an employee/student at the last of the list
b. Find(DelegateMethod): to find an employee/Student from a given list and return the first matched employee/Student.
c. FindAll(DelegateMethod): to find all employee/Student from a given list and return the all matched employee/Student(s).
d. FindIndex(DelegateMethod): to find an employee/Student from a given list and return the index of matched employee/Student.
e. IndexOf(T): to find a T from a given list and return the index of first matched T.
f. LastIndexOf(T): to find a T from a given list and return the last occurrence index of matched T.
g. Remove(T) : to remove an Employee/Student from the given list.
h. RemoveAll(Predicate<T>): to remove all Employees/Students with a given condition.

Assignment Set: J

Objectives of the Assignment:

- Familiarize with LINQ in C# Programming and understanding the power of LINQ in an application.

CO/BTLCovered: CO_1 to CO_3/BTL_2 to BTL_6

Problems:

Classes to be implemented in C# and perform the following actions in classes.

class Student{Roll,Name,Score,City}

class Faculty{ID, Name, Salary, City}

JP$_1$ Retrieve the data from both tables and convert into a List<Student> and List<Faculty>. Where given definition of class Student {Roll, Name, Score, City} and Faculty{ID, Name, Salary, City}.

JP_2 Write a C# Program to display the all Faculties details whose salary is greater than 5000 from a List< Faculty > using LINQ.

JP_3 Write a C# Program to display the all Faculties' and Students' name who lived in same City using LINQ Concat method.

JP_4 Write a C# Program to display the all Faculties and Students details who lived in same City using LINQ join.

JP_5 Write a C# Program to display the List of all students in group as per their score using LINQ.

****