

Laboratory Manual

(Version 10.0)

for

Java Programming Lab

(MCA- 167)

MCA - I Semester

Compiled by:

Dr. Ritika Wason

(Associate 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

Declaration

1. Vision of the Department	5
2. Mission of the Department	5
3. Programme Educational Objectives (PEOs)	5
4. Programme Outcomes (POs)	6
5. Institutional Policy for Students' Conduct	8
6. Learning Outcomes of Laboratory Work	9
7. Course/Lab Outcomes (COs)	10
8. Mapping of COs with POs	10
9. Course/Lab Description	11
10. Grading Policy	11
11. Lesson Plan	12
12. Assignments	15

Appendix- A

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 Environmental Concern

Declaration

Department	:	Department of Computer Science and Applications
Course, Year and the Semester to which Lab is offered	:	MCA I Year, I Semester
Name of the Lab Course	:	Object-Oriented Programming and Java Lab
Course Code	:	MCA-167
Version No.	:	10.0
Name of Course/Lab Teacher	:	Dr. Ritika Wason
Laboratory Manual Committee	:	<ol style="list-style-type: none"> 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. Mrs. Tanya Pathak Garg, Concerned Subject Teacher, Convener
Approved by	:	DAC Date: 03/05/2021
Approved by	:	IQAC Date: 04/05/2019

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

- M1** To become a centre of excellence in the field of Computer Science and Applications and produce professionals as per global industry standards.
- M2** To foster innovation, entrepreneurial skills, research capabilities and bring all-round development amongst budding professionals.
- M3** To promote analytical and collaborative life-long learning skills, among students and faculty members involving all stakeholders.
- M4** 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:

- PEO1** Exhibit professional competencies and knowledge for being a successful technocrat.
- PEO2** Adopt creative and innovative practices to solve real-life complex problems.
- PEO3** Be a lifelong learner and contribute effectively to the betterment of the society.
- PEO4** Be effective and inspiring leader for fellow professionals and face the challenges of the rapidly changing multi-dimensional, contemporary world.

4. Programme Outcomes (POs)

P01: 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.

P02: 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.

P03: 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.

P04: 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.

P05: Modern Tool Usage (MTU)

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

P06: 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.

P07: Life-Long Learning (LLL)

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

P08: 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.

P09: 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.

P010: Societal and Environmental 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.

P011: Individual & Team Work (I&T)

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

P012: 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 Practical File, having prescribed format, as given in Appendix-A. All the students should get the weekly assignment checked and signed in the Practical File by the respective teacher in the immediate succeeding week. The second sets 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 continuous 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)

- CO1** Implement OOPs concepts in Java.(BTL3)
- CO2** Write programs to implement exception handling, multithreading and I/O programming.(BTL4)
- CO3** Build Graphical User Interfaces (GUI) through applets, AWT and swings and devising event handling mechanisms.(BTL5)
- CO4** Create database enabled client-server applications.(BTL6)
- CO5** Work in teams to develop projects for real-life problems using Java.(BTL5&BTL6)

8. Mapping of COs with POs

Table 1: Mapping of COs with POs

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	-	3	3	-	-	-	3	-	-
CO2	3	2	2	-	-	3	-	-	-	3	-	-
CO3	3	3	3	-	2	3	2	-	-	3	-	-
CO4	3	3	3	3	-	3	2	-	-	3	-	-
CO5	3	3	3	2	3	3	2	3	3	3	3	2

9. Course/Lab Description

Course (Lab) Title	: Object-Oriented Programming and Java Lab
Course (Lab) Code	: MCA-167
Credits	: 02
Pre-requisites	: A computer with JDK and Netbeans IDE, basics of object oriented programming concepts, basic logic building skills.
Academic Session	: December to March
Contact Hours/Week	: 04 (02 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 Lab Manual
1.	1.	Implementing the basic Object Oriented Programming concepts in Java using Notepad++ and JDK toolkit.	Assignment A (Problem AP1 To AP4)
	2.	Implementing the concept of Wrapper Classes, Type Casting and auto-boxing and un-boxing.	Assignment A (Problem AP5 To AP9) (Advanced:AA1-AA2)
2.	3.	Developing java applications on the concept of Arrays- single dimension, multi-dimension, ragged arrays.	Assignment B (Problem BP1 To BP4)
	4.	Developing java applications working on complex array arithmetic using Comparable and Comparator interfaces.	Assignment B (Problem BP5 To BP7)
3.	5.	Implementing the concept of inheritance in Java and various types of inheritance available.	Assignment B (Problem BP8 To BP10)
	6.	Buffer Reserved for Revision	Assignment A – B (Advanced:BA1-BA2)
4.	7.	Constructing java programs to see the working of auto-boxing and un-boxing in JSE and the working of Inner Classes in java.	Assignment C (Problem CP1 To CP4)
	8.	Constructing java programs to see the working of Static Inner Classes in java and Exception Handling.	Assignment C (Problem CP5 To CP10)
5.	9.	Buffer Reserved for Revision	Assignment – C (Advanced:CA1-CA3)
	10.	Implementing the concept of Multithreading in Java, practical aspects of concurrency control.	Assignment D (Problem DP1 To DP3)

6.	11.	Creating threads using Thread Class, Runnable Interface and Anonymous Implementations.	Assignment D (Problem DP4 To DP6)
	12.	Familiarizing the concept of block, method and volatile synchronization in Threads and File Handling.	Assignment D (Problem DP7 To DP8)
7.	13.	Connecting machines over the intranet using the concept of TCP and UDP Sockets.	Assignment D (Problem DP9 To DP10)
	14.	Buffer Reserved for Revision	Assignment – D (Advanced:DA1-DA3)
8.	15.	Creating Java Applications for implementing File Handling for reading/writing data from persistent storage and vice-versa.	Assignment E (Problem EP1 To EP3)
	16.	Exploring the Collections Framework and various collection types in Java.	Assignment E (Problem EP4 To EP7)
9.	17.	Buffer Reserved for Revision	Assignment – E (Advanced:EA1-EA3)
	18.	Implementing NetBeans IDE for GUI Development in Java by means of AWT and Swings Framework.	Assignment F (Problem FP1 To FP3)
10.	19.	Introducing event handling model in the same to make intuitive and responsive GUI with the help of NetBeans IDE.	Assignment F (Problem FP4 To FP8)
	20.	Buffer Reserved for Revision	Assignment – F (Problem FP1 to FP8) (Advanced:FA1-FA2)
11.	21.	Connecting Java applications to underlying databases (Apache Derby DB and Oracle 11g) using JDBC API.	Assignment F (Problem FP9 To FP10)
	22.	Exploring Prepared Statement and Callable Statement Interfaces for the database connectivity.	Assignment F (Problem FP11 To FP12)

12.	23.	Buffer Reserved for Revision	Assignment – F (Advanced:FA3)
13.	24.	Constructing RMI client server applications to connect two remote machines for method access.	Assignment G (Problem GP4 To GP6)
	25.	Buffer Reserved for Revision	Assignment – G (Advanced:GA1)

12. Assignments

Assignment Set: A

Objectives of the Assignment:

- Implementing the basic Object Oriented Programming concepts in Java using Notepad++ and JDK toolkit.
- Developing programs to evaluate the concept implementation of Wrapper Classes, Type Casting and Java Libraries.

CO/BTL Covered: CO1/BTL3 & BTL5

Problems:

AP1 Explore the basic java program development scenario in Notepad and cmd by creating an Integer Adder. The adder prints sum of 5 integer numbers without using single variable where input will be taken through command line arguments.

- a) Perform the above code using a function and call it in main().
- b) Make another class and a function in it to perform the above task.

AP2 Develop a Number Reciprocator java application to computes the sum of the reciprocals without using a single variable in the format:

$$1/1 + 1/2 + 1/3 + \dots + 1/10$$

AP3 Demonstrate type conversion in a simple java program by casting and checking output in the following cases:-

- a) Conversion of int to byte
- b) Conversion of double to int
- c) Conversion of double to byte
- d) Conversion of int to char
- e) Conversion of float to short

AP4 Construct a character counter that inputs a piece of text that is analyzed character by character to determine the vowels, spaces and letters used. Fill in the code that computes the number of spaces, vowels, and consonants.

```
public class StringCharacters {

    public static void main(String[] args) {

        String text = "To be or not to be, that is the question;"

        +"Whether this nobler in the mind to suffer"

        +" the slings and arrows of outrageous fortune,"

        +" or to take arms against a sea of troubles,"

        +" and by opposing end them?";

        int spaces = 0, vowels = 0, letters = 0;

        //YOUR CODE HERE

        System.out.println("The text contained vowels: " + vowels + "\n" +
            consonants " + (letters - vowels) + "\n"+ spaces: " + spaces);

    }

}
```

AP5 Construct a number generator to accept three digits (i.e. 0 - 9) and print all its possible combinations. (For example if the three digits are 1, 2, 3 than all possible combinations are: 123, 132,213, 231, 312, 321.)

AP6 A java standalone application makes use of a parameterized method inside a class. Take the following case: Create a class Box and define a method in this class which will return the volume of the box. Initialize two objects for your class and print out the volumes respectively.

AP7 A java standalone application reads in a sentence from the user and prints it out with each word reversed, but with the words and punctuation in the

original order.

AP8 Develop an employee pay generator that works on the following rules-

1. An employee gets paid (hours worked) × (base pay), for each hour up to 40 hours.
2. For every hour over 40, they get overtime = (base pay) × 1.5.
3. The base pay must not be less than the minimum wage (\$8.00 an hour).
4. If it is, print an error. If the number of hours is greater than 60, print an error message.

AP9 A Financial Calculator to calculate the SimpleInterest and CompoundInterest by taking command line values for principal, rate and time.

1. Extend the code to calculate 'Final Value' of investment (V) of an investment (principal P) compounded yearly for T years at interest rate R is given by the formula: $V = P (1 + R)^T$
2. Perform the above code using a function and call it in main().

Make another class and a function in it to perform the above task.

Advanced Problems:

AA1 A Gravity Calculator in java to compute the position of an object after falling for 10 seconds, outputting the position in meters. The formula in Math notation is:

$$x(t) = 0.5 \times at^2 + vit + xi$$

Variable	Meaning	Value
a	Acceleration (m/s ²)	-9.81
t	Time (s)	10
vi	Initial velocity (m/s)	0

xi Initial position 0

Implement the same using:-

1. Using if-else ladder
2. Using Switch case

NOTE- use minimum variables in code and the program should be menu driven that exits only when the user opts to.

AA2 A java application 'MyNumber.java' that performs following operations on a variable 'num' of type double. Code to include the following functionality:

1. Finds the round value of 'num' and stores the result in a variable numRound of type double.
2. Finds the ceil value of 'num' and stores the result in a variable numCeil of type double.
3. Finds the floor value of 'num' and stores the result in a variable numFloor of type double

Cast 'num' to type int and stores the result in a variable numInteger of type int.

Assignment Set: B

Objectives of the Assignment:

- Implement the concept of Arrays- single dimension, multi-dimension, ragged arrays
- Develop java applications working on complex array arithmetic using Comparable and Comparator interfaces.
- Implement the concept of inheritance in Java.

CO/BTL Covered: CO1 & CO2/BTL2 ,BTL3 & BTL5

Problems:

BP1 Develop a command line driven code to accept the following city name as

argument in the command line and sort them in alphabetic order –

City Name = Kolkata, Chennai, Mumbai, Delhi, Bangalore, Ahmedabad

BP2 A group of BVICAM friends decide to run the Airtel Delhi Half Marathon.

Their names and times (in minutes) are below:

Name Time (minutes)

Elena 341

Thomas 273

Hamilton 278

Suzie 329

Phil 445

Matt 402

Alex 388

Emma 275

John 243

James 334

Jane 412

Emily 393

Daniel 299

Neda 343

Aaron 317

Kate 265

Find the fastest runner. Print the name and his/her time (in minutes).

Optional: Find the second fastest runner. Print the name and his/her time (in minutes).

BP3 Create a Waffle Class having two data members flavor and price, being populated by a parameterized constructor. Create another class WaffleMain to compare the values and object context of Waffle class data members by `.equals()` and `==`. Extend the implementation to get desired results by overriding

- a) `equals()`
- b) `toString()`

BP4 Create a POC(Proof of Concept) to demonstrate usage of various functions of String Class like:-

- a) `charAt()`
- b) `length()`
- c) `contains()`
- d) `equals` and `==`
- e) `indexOf()`
- f) `split()`
- g) `toUpperCase()`

BP5 Use ragged array to provide the output given below (Take row count from user).

```
1
123
1234
12345
123456
1234567
1
12
123
1234
12345
123456
1234567
```

BP6 An application maintains information regarding teams and players of a

particular sport. Identify and implement the relation between team and player. Create a program that provides the following options:

- a) Enter TeamID
- b) Enter PlayerID

When the user enters a valid TeamID the details of the players belonging to that team should be displayed PlayerID the details of the team to which the player belongs should be displayed. There can be multiple players associated with a country.

BP7 An Array of Student class objects.

- a) Sort them by roll number using Comparable Interface.
- b) Perform sort based on alphabetic first name numbering using Comparator.

BP8 Code to declare & instantiate a 2D-array to hold marks obtained by students in different subjects in a class. Assume that there are up to 10 students in a class & there are 5 subjects. Find out the best student according to average marks of all subjects and display all the marks of him/her.

BP9 An application maintains information regarding Products and Suppliers for a particular business. Identify and implement the relation between Product and Supplier. Create a program that provides the following options:

- a) Enter ProductID
- b) Enter SupplierID

When the user enters a valid

- a) ProductID the details suppliers who supply that product should be displayed
- b) SupplierID the details of products supplied by that supplier should be displayed

Each product might be supplied by multiple suppliers also one supplier can

supply multiple products.

BP10 Simulate a simple banking application. Provide for classes Bank and interface Account. Account will be of two type- Savings and Current. There should be methods to open an account, close an account and perform withdraw, deposit and transfer operations on an account as abstract methods in Account and properly overridden with definition in Account Types. Test classes should instantiate Account Type Classes and provide a menu driven option for operations.

Advanced Questions:

BA1 Construct a Java program involving two classes: OddAndEven and TestOddAndEven. OddAndEven has the following:

1. Instance variables countOfOdd and countOfEven both of type int
2. A method addNumber that takes a number as parameter and increment countOfOdd, if the number is odd, else increment countOfEven.
3. A method toString that returns a string in the form: "Number of Odd: x, Number of Even : y", where x and y are the values of the instance variables.

The TestOddAndEven class first creates OddAndEven object, then in a loop, read a number and use it to call the addNumber method until the user enters Q. Finally, it prints the count of odd and even numbers entered.

BA2 An Employee Management System has the following classes Employee, PermanentEmployee, TemporaryEmployee such that the class Employee acts as the base class for the other two classes. Add appropriate data members & functions to the base class. The base class should contain an abstract function to calculate net salary.

The class PermanentEmployee should

- a) Contain data members for storing BasicSalary, HRAPercent, DAPercent, PFPercent
- b) Provide body for the abstract function, contained in the base class, to calculate net salary on basis of the data given above.

The class `TemporaryEmployee` should

- a) Contain data members for storing `HourlyWages`, `HoursWorked`
- b) Provide body for the abstract function, contained in the base class, to calculate wages on basis of the data given above.

Test the above classes and demonstrate dynamic binding. Your application should provide options to:

- a) Add data for permanent employee
- b) Add data for temporary employee

Calculate salary of an employee given his/her empno.

Assignment Set: C

Objectives of the Assignment:

- Familiarize with the working of Autoboxing and unboxing in JSE
- Construct java programs to see the working of Inner Classes in java
- Create java applications to see workflow of Exception Handling in Java

CO/BTL Covered: CO1 & CO2/ BTL2, BTL3 & BTL5

Problems:

CP1 Code to demonstrate the use of Autoboxing and Unboxing using:-

- a) Method Passing
- b) Variable Assignment

CP2 A program called **DecipherCaesarCode** to decipher the Caesar's code described in the previous exercise. The program shall prompts user for a ciphertext string consisting of mix-case letters only; compute the plaintext; and print the plaintext in uppercase.

CP3 A word that reads the same backward as forward is called a *palindrome*, e.g., "mom", "dad", "racecar", "madam", and "Radar" (case-insensitive). Create a program called **TestPalindromicWord**, that prompts user for a word and prints ""xxx" is|is not a palindrome".

- CP4** A java based program which prompts user for the number of students in a class (a non-negative integer), and saves it in an int variable called numStudents. It then prompts user for the grade of each of the students (integer between 0 to 100) and saves them in an int array called grades. The program shall then compute and print the average (in double rounded to 2 decimal places) and minimum/maximum (in int).
- CP5** Demonstrate the working of a Static Inner Class through a class Electronics and within it create Static Inner Class Television that has a method cost() which displays cost of television object passed in constructor of Television class. Demonstrate invoking inner class method with outer object when the method cost() is once a :-
- a) Instance(Non static) method
 - b) Static method
- CP6** Demonstrate the working of Anonymous Inner Class in any utility of your project through:-
- a) Anonymous Inner Sub Class implementation
 - b) Anonymous Inner Interface implementation
- CP7** Extend your Gravity Calculator code (Assignment 1) to handle exceptions through a try-catch finally block. Handle provision for a divide by zero scenarios caught by NumberFormatException. Explicitly invoke this exception in execution and observe the response.
- CP8** Further extend your menu-driven Simple Calculator code (Assignment 2) by adding support a Custom Exception code that raises an 'Invalid Numeral' exception each time the user tries to enter any character except a number for calculation. Explicitly invoke this exception in execution and observe the response.
- CP9** An ExceptionPOC class is requesting a number between 1 and 10. Run the program again and enter 5.5. Although this number is between 1 and 10, the

program will abort. Examine the error message. You should see the word Exception, the method where the exception occurred (main), the class name of the exception (InputMismatchException), as well as the call stack listing the method calls.

Add a try/catch block to catch and handle the InputMismatchException exception. Identify the statements that cause the error as well as the portions of the program that depend upon these statements. Enclose these statements within the try block. Follow the try block with the catch block given below. Note, the InputMismatchException class is defined in java.util and must be imported. Also, when the Scanner throws an InputMismatchException, the input token will remain in the buffer so that it can be examined by the program.

Complete code by implementing the same using:-

- a) Throws method declaration
- b) Throw keyword

CP10 Extend the following methods in a class called DateUtil:

- a) boolean isLeapYear(int year): returns true if the given year is a leap year. A year is a leap year if it is divisible by 4 but not by 100, or it is divisible by 400.
- b) boolean isValidDate(int year, int month, int day): returns true if the given year, month and day constitute a given date. Assume that year is between 1 and 9999, month is between 1 (Jan) to 12 (Dec) and day shall be between 1 and 28|29|30|31 depending on the month and whether it is a leap year.
- c) int getDayOfWeek(int year, int month, int day): returns the day of the week, where 0 for SUN, 1 for MON, ..., 6 for SAT, for the given date. Assume that the date is valid.

String toString(int year, int month, int day): prints the given date in the format "xxxd day d mmm yyyy", e.g., "Tuesday 14 Feb 2012". Assume that the

given date is valid.

Advanced Questions:

CA1 Design a Banking Application in java to represent a bank account. Include the following members.

Data members

- a) Name of depositor
- b) Account Number
- c) Type of account
- d) Balance account in the account

Methods

- a) To assign initial values
- b) To deposit an account
- c) To withdraw an account after checking balance.
- d) To display the name and balance

CA2 A program called Grades Statistics, which reads in n grades (of int between 0 and 100, inclusive) and displays the average, minimum, maximum, median and standard deviation. Display the floating-point values upto 2 decimal places.

CA3 A positive integer is called a perfect number if the sum of all its factors (excluding the number itself, i.e., proper divisor) is equal to its value. For example, the number 6 is perfect because its proper divisors are 1, 2, and 3, and $6=1+2+3$; but the number 10 is not perfect because its proper divisors are 1, 2, and 5, and $10 \neq 1+2+5$.

A positive integer is called a deficient number if the sum of all its proper divisors is less than its value. For example, 10 is a deficient number because $1+2+5 < 10$; while 12 is not because $1+2+3+4+6 > 12$.

Write a boolean method called `isPerfect(int aPosInt)` that takes a positive integer, and return true if the number is perfect. Similarly, write a boolean method called `isDeficient(int aPosInt)` to check for deficient numbers.

```
public static boolean isPerfect(int aPosInt);
```

```
public static boolean isDeficient(int aPosInt);
```

Using the methods, write a program called PerfectNumberList that prompts user for an upper bound (a positive integer), and lists all the perfect numbers less than or equal to this upper bound. It shall also list all the numbers that are neither deficient nor perfect.

Assignment Set: D

Objectives of the Assignment:

- Implement the concept of Multithreading in Java. Consider practical aspects of concurrency control and synchronization in the same.
- Connect machines over the intranet using the concept of TCP and UDP Sockets.

CO/BTL Covered: CO2 & CO3/BTL2 , BTL3 & BTL4

Problems:

DP1 A PalindromesViaThreads class that prints the prints 20 palindrome numbers(121,131,232) between 100 and 1000 using Threads implemented via:-

- a) Extending Thread Class
- b) Implementing Runnable Interface

DP2 Extend your Banking Application exercise (Assignment 4, Ques 5, Unit 1) to include the concept of Synchronization in a multithreaded environment using the approach of block synchronization. Methods to be made thread safe are:-

- a) withdrawl()
- b) deposit()

DP3 A program called ReverseHello creates a thread (let's call it Thread 1). Thread 1 creates another thread (Thread 2); Thread 2 creates Thread 3; and so on, up to Thread 50. Each thread should print "Hello from Thread <num>!", but you should structure your program such that the threads print their greetings in reverse order.

DP4 A SharedCounter has 10 threads each increment a shared int counter variable 10 times. When all the threads have finished, print the final value of the counter. If the initial value is zero, do you always get 100? Arrange for your code to sometimes print the wrong answer. (Hint: try using some well-placed calls to Thread.yield() or Thread.sleep())

DP5 Compile and run BadThreads.java:

```
public class BadThreads {
    static String message;
    private static class CorrectorThread
        extends Thread {
        public void run()
            {try {
                sleep(1000);
            } catch (InterruptedException e) {}
            // Key statement 1:
            message = "Mares do eat oats.";
        }}
    public static void main(String args[])
        throws InterruptedException {
        (new CorrectorThread()).start();
        message = "Mares do not eat oats.";
        Thread.sleep(2000);
        // Key statement 2:
        System.out.println(message);
    }}
}
```

The application should print out "Mares do eat oats." Is it guaranteed to always do this? If not, why not? Would it help to change the parameters of the two invocations of Sleep? How would you guarantee that all changes to message will be visible in the main thread?

DP6 Create a File Shredding application that shreds a given file where filename and the size of shredded file are given as command line arguments. Test that the file was shredded properly using the DOS command:

Java Copy a b c ... dest

Where a, b, c etc are the source files and dest is the destination file

Also, write a java program to combine the files generated above into a single file.

DP7 Construct a Java Application that takes a flag and a file name form the user as command line parameters. The program should provide the following features:

Flag	Action
-cc	Count the number of characters in the file
-cw	Count the number of words in the file
-cl	Count the number of lines in the file
-s	Search the specified string in the file and display all the lines along with the line numbers where the string appears. (will require additional command line parameter for the word to be searched)

DP8 A connection-oriented client/server application using TCP/IP protocol where the client that has the following responsibilities: a. Server: Creates an Employee Class having fields- employeeName, employeeID and department.

- a) Server holds an array of employee objects.
- b) Client: Accept the employeeID of an employee as an integer from the

user.

- c) Server: Searches for corresponding employee object, in the array and write its details to the client stream.
- d) Client: Display the received object's information.

DP9 Create a connection-less Client/Server application using UDP protocol that sends system date and time in the format requested by the client.

- a) Client: Reads a string representing the required format from the end-user.
- b) Server: Returns the system date and time in the requested format or a default format if received format is not understandable.
- c) Client: Display the returned contents

DP10 Construct a connection-less TimeServer application that sends system date and time in the format requested by the client.

- a) Client: Reads a string representing the required format from the end-user.
- b) Server: Returns the system date and time in the requested format or a default format if received format is not understandable.
- c) Client: Display the returned contents

Advanced Problems:

DA1 Implement the producer consumer problem using multi-threading in java.

In computing, the producer-consumer problem (also known as the bounded-buffer problem) is a classic example of a multi-process synchronization problem. The problem describes two processes, the producer and the consumer, which share a common, fixed-size buffer used as a queue.

- a) The producer's job is to generate data, put it into the buffer, and start again.
- b) At the same time, the consumer is consuming the data (i.e. removing it

from the buffer), one piece at a time.

To make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer

DA2 Create a real-time chatbot application by extending classroom examples on one sided communication on TCP and UDP. Create the application using two protocols:

- a) Connection-oriented TCP/IP
- b) Connection-less UDP

DA3 Construct a connection-less dictionary lookup Client-Server application that has following features:

- a) Client: Reads a string from the end-user.
- b) Server: Looks up the word in a dictionary (maintained as a text file) containing name-meaning pairs and sends the meaning back to the client.
- c) Client: Display the returned contents

Assignment Set: E

Objectives of the Assignment:

- Implement File Handling for reading/writing data from persistent storage and vice-versa.
- Explore the Collections Framework and various collection types in Java.

CO/BTL Covered: CO1 & CO3/BTL3 & BTL6

Problems:

EP1 Extend your String tokenizer application(Lab Assignment 4, Ques 1, Unit 1) to store the input string contents in a file. Read the file and count vowels, consonants and spaces in each line. Create another file to write the vowel and consonant count besides each line. For eg:-

Hi this is java(vowels-5, consonants-7, spaces- 3). I like studying it(vowels-6,

consonants-9, spaces-4).

Perform this operation using:

- a) BufferedReader and BufferedWriter
- b) FileReader and FileWriter

EP2 A result generator application maintains objects of class Result in a TreeSet. The ordering of objects is to be based on Semester and then roll no. The class has following members:

- a) Rollno: String
- b) Semester: String
- c) Array for marks in five subjects

The program should produce the result per semester of roll-number sorted students in the following format:

RollNo: -----

SEM MARKS1 MARKS2 MARKS3 MARKS4 MARKS5 Sum

Total: -----

EP3 Create an xml configurator which is a parsed XML file storing user login credentials and database credentials. Parse this file into java code using DOM parser/SAX Parser. Create Properties in java using these configurations. Retrieve all config values using properties methods.

EP4 Extend your Student Class (Assignment 3, Ques 3, Unit 1) to Serialize objects of student class into separate files and byte streams using Serializable interface. Make use of the serialVersionUID field and declare few variables as transient. Deserialize the objects and store them into an array of objects on another JVM instance and perform sorting based on parameter of user's choice using Comparator.

Note: You will have to make separate comparator classes for different data member based sorting.

EP5 Same as in question 4, Serialize your Student Class using the Externalizable interface overriding the writeExternal() and readExternal() methods. Deserialize the objects and store them into an array of objects on another JVM instance and perform sorting based on parameter of user's choice using Comparator.

Note: You will have to make separate comparator classes for different data member based sorting.

EP6 A File Parser a file and store the following text in it-

“Dwelling and speedily ignorant any steepest. Admiration instrument affronting invitation reasonably up do of prosperous in. Shy saw declared age debating ecstatic man. Call in so want pure rank am dear were. Remarkably to continuing in surrounded diminution on. In unfeeling existence objection immediate repulsive on he in. Imprudence comparison uncommonly me he difficulty diminution resolution. Likewise proposal differed scarcely dwelling as on raillery. September few dependent extremity own continued and ten prevailed attending. Early to weeks we could.

Unpleasant astonished an diminution up partiality. Noisy an their of meant. Death means up civil do an offer wound of. Called square an in afraid direct. Resolution diminution conviction so mr at unpleasing simplicity no. No it as breakfast up conveying earnestly immediate principle. Him son disposed produced humoured overcame she bachelor improved. Studied however out wishing but inhabit fortune windows.”

Accept a SearchToken from the user. Open the file and read it using RandomFileAccess and search and display total occurrences of the search string in given text.

EP7 A Student class modeling the basic data members of entity student. Set up getters and setters for each data member. Create another class that converts the object of Student class to byte stream using ObjectOutputStream on one JVM(command prompt instance). Open up another JVM and read the byte stream of object from the same file. Recreate the object from byte stream on

2nd JVM using ObjectOutputStream and display its content.

Advanced Problems:

EA1 A java application maintains 5 objects of class Employee in a HashSet. The class has following members:

- (a) Empno: String
- (b) Department: String
- (c) Basic: int
- (d) HRA_Percent: int
- (e) DA_Percent: int
- (f) PF_Percent: int

Sort and segregate the objects on the basis of Department field. Print the sorted result department wise in the following format:

Department: -----

EMPNO BASIC HRA DA PF NET_SALARY

.....-.....

.....-.....

.....-.....

Department Average: -----

EA2 A class Student having data members stu_id, stu_name, stu_course. Create 10 objects of this class as desired by the user. Store each object in a sorted order in a hashmap with key as stu_id and value as student object. Create a menu driven code, to provide user options to:

- a) Add student object in hashmap
- b) Retrieve any desired object based on roll number entered
- c) Update details of student object in the map
- d) Remove student object from map

Sort and copy objects in another hashmap

EA3 A file handler to maintain file using `DataInputStream` / `DataOutputStream` built on top of respective file streams. The file is to store objects of the class `Course` containing the following members:

CourseID: String (Key)

CourseName: String

Fee: float

Implement menu driven program with following. Options

- a) Append
- b) Searching of Course using CourseID

Assignment Set: F

Objectives of the Assignment:

- Familiarizing the GUI Development in Java by means of AWT and Swings Framework.
- Implement event handling model in the same to make intuitive and responsive GUI with the help of NetBeans IDE.
- Connecting Java applications to underlying databases using JDBC API. Exploring Prepared Statement and Callable Statement Interfaces for the same.

CO/BTL Covered: CO4 & CO5/BTL5 & BTL6

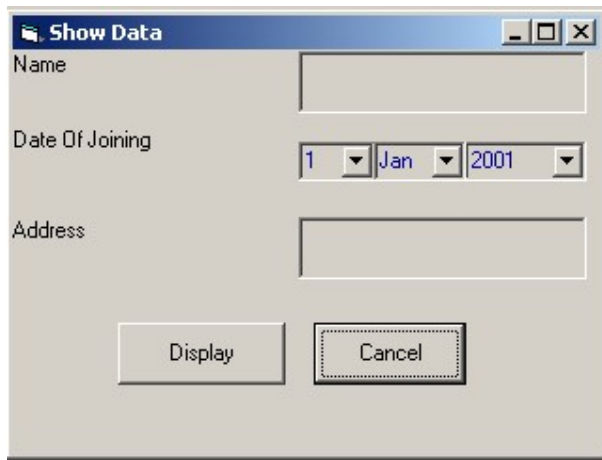
Problems:

FP1 Use AWT /Swings to develop a Contact Manager. The GUI application

maintains a simple list of contacts storing contact name and number. It provides basic features of CRUD (Create, Update, Retrieve and Delete) as per user choice. The application also supports search feature.

FP2 Create a class `MessageBox` that extends either `Dialog` or `JDialog`. The class should have a constructor that takes a `String` as a parameter to construct a dialog box that displays the message and `OK` & `CANCEL` buttons. The dialog box should get closed when the cancel buttons is clicked. Provide some mechanism in the `MessageBox` class that can be used by the calling program to check which button was pressed by the user.

FP3 Create the following layout using `awt` / `swing`



When user clicks the “Display” button the data entered by the user should be displayed in another frame window.

When user clicks the “Clear” button the data fields should get cleared.

FP4 Create a `DALWrapper` class for your project where you test basic `JDBC` connectivity in your application. Perform basic `CRUD` from database in your application. Make use of the following concepts in your code:

- a) `PreparedStatement`
- b) `CallableStatement`

FP5 Create a class `InputDialog` that extends `Dialog` / `JDialog`. The class should have a constructor that takes a `String` as a parameter to construct a dialog box that

displays a text box (whose initial contents are specified by the string parameter) and OK & CANCEL buttons. The dialog box should get closed when either of the buttons is clicked. The class should have functions to:

- a) Check which button was pressed by the user.
- b) Retrieve the string entered by the user, if user pressed OK, null if user pressed CANCEL.

Test this class to get a message from a user and display it on a Frame window.

FP6 Develop a panel that simulates MessageRotator component. The panel should use multithreading to pick up messages from an ArrayList and display the messages one by one. The messages are set in the constructor.

FP7 Create a frame that displays a text area in center and text filed in the south region of the frame. The text area is supposed to hold contents of a text file.

The text field is supposed to hold a file name.

The frame should have a menu with the following options:

File

New

Open

Save

Exit

When user clicks New both the input fields should be cleared.

When user clicks Open the contents of the text file whose name is in text field should be displayed in the text area.

When user clicks Save the contents of the text area should get saved in the file whose name is given in the text field.

Clicking Exit should terminate the application.

FP8 Create a frame that can be used to perform simple drawing operations. The

frame should have following menu options:

Shape	Style	Color
Line	Fill	Red
Rectangle	Draw	Blue
Oval		Green

The point where the user clicks the left mouse button should be treated as the top-left corner of the shape and the point where the user releases the left mouse button should be treated as the bottom-right corner of the shape.

FP9 Create a frame that displays a pie chart based upon a set of numerical-values entered by the user as command line argument.

FP10 Create a frame that displays a bar chart based upon a set of numerical values entered by the user as command line argument.

FP11 A Java application that uses JDBC to connect to a database containing table that holds data for students of your class. The application should allow the user to:

- a) Add a record
- b) Search for a record
- c) Modify an existing record

Make use of Prepared Statement

Extend the application to call a backend procedure/function. The procedure should take ID field as input parameters and return details of related record.

Advanced Problems:

FA1 Create a class called ColorDialog that allows user to choose a color. The dialog should display three text boxes where a number in the range 0-255 can

be entered. The dialog box should further have three buttons: Test, Ok and Cancel. When user clicks:

- a) Test: The entered values should be used to create a color and the color should be set as the background color of a test label.
- b) Ok: The selected color should be saved in a class member for further reference and the dialog should get closed.
- c) Cancel: The dialog should get closed without saving the entered values.

FA2 Use the following layout to navigate through a set of objects stored in an ArrayList. The objects should have the members as indicated by the input fields:



The image shows a Java Swing dialog box titled "Navigate". It has a standard window title bar with minimize, maximize, and close buttons. The dialog contains three input fields: "Name", "City", and "Hobbies". The "Hobbies" field has two checkboxes: "Playing" (unchecked) and "Reading" (checked). At the bottom of the dialog, there are four buttons: "First", "Previous", "Next", and "Last".

When user clicks "Previous" when the first record is displayed, the application should navigate to Last record, similarly when user clicks "Next" when the last record is displayed, the application should navigate to the First record.

FA3 Use the following layout to navigate through a set of objects stored in an ArrayList. The objects should have the members as indicated by the input fields:



When user clicks “Previous” when the first record is displayed, the application should navigate to Last record, similarly when user clicks “Next” when the last record is displayed, the application should navigate to the First record.

Assignment Set: G

Objectives of the Assignment:

- Construct RMI client server applications to connect two remote machines for method access.

CO/BTL Covered: CO3, CO4 & CO5/ BTL4, BTL5 & BTL6

Problems:

GP1 Create an RMI application that exposes a remote object School. It should have two remote methods admit and search. The method admit should add a student’s record in the list of available students and search should return the details of student on basis of roll number entered or raise an exception in case of invalid roll number. Demonstrate the use of these methods in a RMI client application.

following concepts:

- a) Methods class
- b) Field class

GP2 Implement a RMI application. The server object maintains a HashMap of Faculty and Course names; and provides for a method subjectTaught that takes a faculty name as parameter. The client should attain a remote reference to the object and call the remote method to satisfy user queries.

GP3 An RMI application that exposes a remote object DepartmentalStore. It should have two remote methods isAvailable and purchase. The method isAvailable should check the availability of a product and purchase should modify the stock of the product to be purchased. Demonstrate the use of these methods in a RMI client application.

Advanced Problems:

GA1 Implement a RMI application. The server object maintains a HashMap of Department and Employee names and provides a remote method getEmployeeNames that takes a department name as parameter. The client should attain a remote reference to the remote object and call the remote method to satisfy user queries.

APPENDIX-A**Template for the Index of Lab File**

WEEK NO.	PROBLEMS WITH DESCRIPTION		PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
1	AP1			
	AP2			
	AP3			
2	AA1			
	AA2			
	AA3			
3	BP1			
	BP2			
	BP3			
	BP4			

Note: The students should use Header and Footer mentioning their roll no. & name in header and page no. in footer