Diploma Engineering

Laboratory Manual

Object Oriented Programming
With JAVA
(4341602)

4th Sem. - Information Technology

Enrolment No	
Name	
Branch	
Academic Term	
Institute	

Directorate of Technical Education
Gandhinagar -Gujarat.

DTE's Vision:

- To provide globally competitive technical education;
- Remove geographical imbalances and inconsistencies;
- Develop student friendly resources with a special focus on girls' education and support to weaker sections;
- Develop programs relevant to industry and create a vibrant pool of technicalprofessionals.

Institute's Vision:

Institute's Mission:

Department's Vision:

Department's Mission:

Certificate

This is to certify that Mr./Ms.....

Enrollment No. .....of..... Semester of Diploma

in..... of

Institute,.....

.....

(GTU Code: .....) has satisfactorily completed the term worksatisfactorily in course **Object Oriented Programming with JAVA –434002** for the academic year:..... Term: ..... as prescribed in the GTU curriculum.

Place: .....

Date: .....

Signature of Course Faculty Head of the Department

Preface

The primary aimof any laboratory/Practical/field work is for enhancing required skills as well as creating ability amongst students to solve real time problem by developing relevant competencies in psychomotor domain. By keeping in view, GTU has designed competency focused outcome-based curriculum-2021 (COGC-2021) for engineering diploma programmers. In this more time allotted to practical work than theory. It shows importance of enhancement of skills amongst students and it pays attention to utilize every seconds of time allotted for practical amongst students, instructors and Lecturers to achieve relevant outcomes by performing rather than writing practice in study type. It is essential for effective implementation of competency focused outcome-based green curriculum-2021, every practical has been keenly designed to serve as a tool to develop & enhance relevant industry needed competency in each and every student. These psychomotor skills are very difficult to develop through traditional chalk and board content delivery method in the classroom. Accordingly, this lab manual designed to focus on the industry defined relevant outcomes, rather than old practice of conducting practical to prove concept and theory.

By using this lab manual, students can read procedure one day in advance to actual performance day of practical experiment which generates interest and also, they have idea of judgement of magnitude prior to performance. This in turn enhances predetermined outcomes amongst students. Each and every experiment /Practical in this manual begins by competency, industry relevant skills, course outcomes as well as practical outcomes which serve as a key role for doing the practical. The students will also have a clear idea of safety and necessary precautions to be taken while performing practical.

This manual also provides guidelines to lecturers to facilitate student-centered lab activities through each practical / experiment by arranging and managing necessary resources in order that the students follow the procedures with required safety and necessary precautions to achieve outcomes. It also gives an idea that how students will be assessed by providing Rubrics.

Object oriented programming with JAVA course develop a depth understating about programming paradigm, object orients programming concept like Inheritance, Encapsulation, Abstraction, polymorphism etc. In the current industrial scenario JAVA has broad industry support and prerequisite to many technologies. After learning this course student can write/develop platform independent, object-oriented application using JAVA.

Although we try our level best to design this lab manual, but always there are chances of improvement. We welcome any suggestions for improvement.

Program Outcomes (Pos):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science andengineering fundamentals and engineering specialization to solve the engineering problems.

2. **Problem analysis:** Identify and analyze well-defined engineering problems using codifiedstandard methods.

3. **Design/ development of solutions:** Design solutions for engineering well-defined technicalproblems and assist with the design of systems components or processes to meet specified needs.

4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools andappropriate technique to conduct standard tests and measurements.

5. **Engineering practices for society, sustainability and environment:** Apply appropriatetechnology in context of society, sustainability, environment and ethical practices.

6. **Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

7. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

Practical Outcome - Course Outcome matrix

NO	PRACTICAL OUTCOME	CO1	CO2	CO3	CO4
1	Install JDK and Setup a Java Programming development environment by using: Command Prompt (SET PATH command and using Environment Variable). Any open source IDE	√	-	-	-
2	Test the java development environment setup by implementing a simple java program (print: “OOP with JAVA”).	√	-	-	-
3	Develop a basic java program that demonstrates data types of JAVA.	√	-	-	-
4	Develop a Java program to swap two numbers without using a temporary variable and with using a temporary variable (use command line argument to accept value from user).	√	-	-	-
5	Develop programs to demonstrate use of - if statement and its different form switch case statement	√	-	-	-
6	Develop program to demonstrate use of- for loop ‘while’ and ‘do while’ loop	√	-	-	-
7	Develop a Java program to find maximum and minimum numbers from array elements.	√	-	-	-
8	Develop a basic java program that demonstrates the use of Class & Object.	-	√	-	-
9	Develop a java program to find the factorial of a given number using a recursive function.	-	√	-	-
10	Develop a java program that demonstrates method overloading.	-	√	-	-
11	Develop a program for implementation of different functions of String class.	-	√	-	-
12	Develop a program for implementation of Wrapper Class to convert primitive value into object (Boxing) and object into primitive value (Un-boxing).	-	√	-	-
13	Develop a program with a static block and show that it will be executed before the main() method in a class.	-	√	-	-
14	Develop a program to demonstrate use of static functions.	-	√	-	-
15	Develop a program to demonstrate use of ‘this’ keyword. Check whether ‘this’ can access the private members of the class or not.	-	√	-	-
16	Develop a program with an overloaded constructor. Also develop the copy constructor to create a new object with the state of the existing object.	-	√	-	-

17	Develop a program to demonstrate the use of private constructor and also write a method which will count the number of instances created using default constructor only.	-	√	-	-
NO	**PRACTICAL OUTCOME**	**CO1**	**CO2**	**CO3**	**CO4**
18	Develop a program to demonstrate single inheritance, multilevel inheritance, and hierarchical inheritance.	-	√	-	-
19	Develop a program with one class named shape which has two member functions named erase () and draw(). In the program we have three other subclasses: circle, triangle and square. Override methods of the super class into sub classes.	-	√	-	-
20	Develop a program for implementation of Dynamic method dispatch.	-	√	-	-
21	Develop a java program that demonstrates the use of Abstract class.	-	√	-	-
22	Develop a java program that illustrates interface inheritance. Interface 'A1' and 'A2' are extended from interface 'A'. Interface 'A12' inherited from both 'A1' and 'A2'. Each interface declares one method and one constant. Class 'Interface_Imple' implements 'A12'. Instantiate 'Interface_Imple' and invoke each of its methods. Each method displays one of the constants.	-	√	-	-
23	Develop a program to create a Package and demonstrate how packages are used in java. and use java access modifier to demonstrate the access rules in a package.	-	√	-	-
24	Develop a program to demonstrate the use of 'super' and 'final' keywords.	-	√	-	-
25	Develop programs to demonstrate the use of Exception Handling using predefined Exception Classes.	-	-	√	-
26	Develop a program to handle multiple exceptions using multiple try blocks and multiple catch blocks.	-	-	√	-
27	Develop a program to implement user defined exceptions.	-	-	√	-
28	Develop a program to demonstrate use of throw, throws, and finally keyword.	-	-	√	-
29	Develop a program that executes two threads. One thread displays "Java Programming" every 2 seconds, and the other displays "Semester - 4th" every 5 seconds. (Create the threads by extending the Thread class)	-	-	√	-
30	Develop a program that executes two threads. One thread will print the even numbers and another thread will print odd numbers between 1 to 10. (Create the thread by implementing runnable interface)	-	-	√	-
31	Develop a program to demonstrate use of synchronization of threads when multiple threads are trying to update a common variable.	-	-	√	-
32	Develop programs to create, write, modify, read operations on Text files.	-	-	-	√

Industry Relevant Skills

The following industry relevant skills of the competency "Develop a program using object-oriented programming concepts with java to solve the given problems." are expected to be developed in the student by undertaking the practical of this laboratory manual.

1. Select the appropriate programming paradigm to solve real world problems.
2. Develop object-oriented programs to support ideas of reusability, abstraction, encapsulation, and polymorphism.
3. Develop a package with library classes.
4. Perform file operation on text file.

Guidelines to Teachers

1. Course faculty should demonstrate experiment with all necessary implementationstrategies described in curriculum.
2. Course faculty should explain industrial relevance before starting of eachexperiment.
3. Course faculty should involve& give opportunity to all students for hands onexperience.
4. Course faculty should ensure mentioned skills are developed in the students byasking.
5. Utilize 2 hrs. of lab hours effectively and ensure completion of write up with quiz also.
6. Encourage peer to peer learning by doing same experiment through fast learners.

Instructions for Students

1. Organize the work in the group and make record of all observations.
2. Students shall develop maintenance skill as expected by industries.
3. Student shall attempt to develop related hand-on skills and build confidence.
4. Student shall develop the habits of evolving more ideas, innovations, skills etc.
5. Student shall refer technical magazines and data books.
6. Student should develop habit to submit the practical on date and time.
7. Student should well prepare while submitting write-up of exercise.

Index

(Continuous Assessment Sheet)

Enrolment No: **Term:**

Name:

NO	EXPERIMENT / PRACTICAL OUTCOME	Page	Date	Marks(25)	Sign
1	Install JDK and Setup a Java Programming development environment by using: 1. Command Prompt (SET PATH command and using Environment Variable). 2. Any open source IDE				
2	Test the java development environment setup by implementing a simple java program (print: “OOP with JAVA”).				
3	Develop a basic java program that demonstrates data types of JAVA.				
4	Develop a Java program to swap two numbers without using a temporary variable and with using a temporary variable (use command line argument to accept value from user).				
5	Develop programs to demonstrate use of if statement and its different formswitch case statement				
6	Develop program to demonstrate use offor loop‘while’ and ‘do while’ loop				

7	Develop a Java program to find maximum and minimum numbers from array elements.				
NO	**EXPERIMENT / PRACTICAL OUTCOME**	**Page**	**Date**	**Marks(25)**	**Sign**
8	Develop a basic java program that demonstrates the use of Class & Object.				
9	Develop a java program to find the factorial of a given number using a recursive function.				
10	Develop a java program that demonstrates method overloading.				
11	Develop a program for implementation of different functions of String class.				
12	Develop a program for implementation of Wrapper Class to convert primitive value into object (Boxing) and object into primitive value (Un-boxing).				
13	Develop a program with a static block and show that it will be executed before the main() method in a class.				
14	Develop a program to demonstrate use of static functions.				
15	Develop a program to demonstrate use of 'this' keyword. Check whether 'this' can access the private members of the class or not.				

16	Develop a program with an overloaded constructor. Also develop the copy constructor to create a new object with the state of the existing object.				
NO	**EXPERIMENT / PRACTICAL OUTCOME**	**Page**	**Date**	**Marks(25)**	**Sign**
17	Develop a program to demonstrate the use of private constructor and also write a method which will count the number of instances created using default constructor only.				
18	Develop a program to demonstrate single inheritance, multilevel inheritance, and hierarchical inheritance.				
19	Develop a program with one class named shape which has two member functions named erase () and draw(). In the program we have three other subclasses: circle, triangle and square. Override methods of the super class into sub classes.				
20	Develop a program for implementation of Dynamic method dispatch.				
21	Develop a java program that demonstrates the use of Abstract class.				
22	Develop a java program that illustrates interface inheritance. Interface 'A1' and 'A2' are extended from interface 'A'. Interface 'A12' inherited from both 'A1' and 'A2'. Each interface declares one method and one constant. Class 'Interface_Imple' implements 'A12'. Instantiate 'Interface_Imple' and invoke each of its methods. Each method displays one of the constants.				
23	Develop a program to create a Package and demonstrate how packages are used in java. and use java access modifier to demonstrate the access rules in a package.				
24	Develop a program to demonstrate the use of 'super' and 'final' keywords.				

25	Develop programs to demonstrate the use of Exception Handling using predefined Exception Classes.				
NO	**EXPERIMENT / PRACTICAL OUTCOME**	**Page**	**Date**	**Marks(25)**	**Sign**
26	Develop a program to handle multiple exceptions using multiple try blocks and multiple catch blocks.				
27	Develop a program to implement user defined exceptions.				
28	Develop a program to demonstrate use of throw, throws, and finally keyword.				
29	Develop a program that executes two threads. One thread displays “Java Programming” every 2 seconds, and the other displays “Semester - 4th” every 5 seconds. (Create the threads by extending the Thread class)				
30	Develop a program that executes two threads. One thread will print the even numbers and another thread will print odd numbers between 1 to 10. (Create the thread by implementing runnable interface)				
31	Develop a program to demonstrate use of synchronization of threads when multiple threads are trying to update a common variable.				
32	Develop programs to create, write, modify, read operations on Text files.				

Practical No:1	**Install JDK and Setup a Java Programming development environment by using:** **1) Command Prompt (SET PATH command and using Environment Variable).** **2) Any open source IDE**

A. Objective:

Configure java development environment by installing appropriate version of JDK.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Engineering Tools, Experimentation and Testing**: Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Identify, download and installation of appropriate JDK (Java Development Kit) for the computer or laptop.
- Setup of java environment path (using SET PATH command and environment variable)
- Setup and use IDE (Integrated Development Environment) to develop java programs.

D. Expected CourseOutcomes (COs):

CO-1: Understand OOP (Object-Oriented Programming) concepts with java.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Identify, download and install appropriate JDK (Java Development Kit) for the development computer, laptop etc.
2. Setup java environment path (using SET PATH command and environment variable)
3. Setup and use IDE (Integrated Development Environment) to develop java programs.

F. Relevant Affective domain related Outcome (ADOs)

1. Handle computer systems carefully with safety and necessary precaution.
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software

development environment which is used to develop java applications and applets. It physically exists. It contains JRE + development tools.

JDK is an implementation of any one of the below given Java Platforms released by Oracle corporation:

- Standard Edition Java Platform
- Enterprise Edition Java Platform
- Micro Edition Java Platform

The JDK contains a Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) etc. to complete the development of a Java Application.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical relatedQuiz/Questions.

1. Write full-form: 1) JDK 2) JVM 3) JRE
2. Compare JDK with JRE.
3. Write five-names of IDEs used to develop java applications.

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

N. References / Suggestions

1. https://www.youtube.com/watch?v=pqqY9jakmFw
2. https://www.youtube.com/watch?v=hQyTvkndyPQ
3. https://www.geeksforgeeks.org/setting-environment-java/
4. https://www.tutorialspoint.com/java/java_environment_setup.htm
5. https://www.javatpoint.com/javafx-how-to-install-java

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practicalhim/herself	Performed Practical with others help	Watchedotherstudents performingpracticalbut nottriedhim/herself	Presenting practicalsessionbutnotattentivelyparticipatedinperformance
Accuracy	5	Accurately done	1-2Errors/mistakesfound	3-5Errors/mistakesidentified	Morethan5Errors /mistakescommitted
Documentation	5	Noerrors,Programiswell ExecutedandDocumentedProperly.	Completewrite-upandoutputtablesbutpresentationispoor	Someofthecommands missingwithmissingoutputs	Poorwrite-upanddiagramormissingcontent
Understanding& Explanation	5	Fullyunderstoodthe performance&canexplainperfectly	Understood theperformance but cannot explain	Partiallyunderstood theperformance&cangivelittleexplanation	Partiallyunderstood andcannotgiveexplanation
Time	5	Completedtheworkwithin1week	Workissubmittedlaterthan1st weekbutbytheendof2ndweek	Workdoneafter2ndweekbutbefore the end of3rdweek	Worksubmittedafter3rd weektime

Sign with Date

Practical No:2	Test the java development environment setup by implementing a simple java program (print: "OOP with JAVA").

A. Objective:

Implement a simple java program and test the java development environment by executing this java program.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills.

D. Expected CourseOutcomes (COs):

CO-1: Understand OOP (Object-Oriented Programming) concepts with java.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Develop and run a simple java program.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

To execute any Java program, the following software or application must be properly installed.

- Install the JDK if you don't have installed it, download the JDK and install it.
- Set path of the jdk/bin directory.
- Create the Java program
- To compile the Example program, execute the compiler, javac, specifying the name of thesource file on the command line, as shown here:

```
C:\>javac Example.java
```

- The javac compiler creates a file called Example.class that contains the bytecode version of the program. Java bytecode is the intermediate representation ofyour program that contains instructions the Java Virtual Machine will execute. Thus, theoutput of javac is not code that can be directly executed.
- To actually run the program, you must use the Java application launcher called java. To do so, pass the class name Example as a command-line argument, as shown here:

```
C:\>java Example
```

Parameters used in First Java Program

- **class** keyword is used to declare a class in Java.
- **public** keyword is an access modifier that represents visibility. It means it is visible to all.
- **static** is a keyword. If we declare any method as static, it is known as the static method. The core advantage of the static method is that there is no need to create an object to invoke the static method. The main() method is executed by the JVM, so it doesn't require creating an object to invoke the main() method.
- **void** is the return type of the method. It means it doesn't return any value.
- **main** represents the starting point of the program.
- **String[] args** or String args[] is used for command line argument.
- **System.out.println()** is used to print statement. Here, System is a class, out is an object of the PrintStream class, println() is a method of the PrintStream class.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

NIL

N. References / Suggestions

1. https://www.javatpoint.com/simple-program-of-java

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No:3	Develop a basic java program that demonstrates data types of JAVA.

A. Objective:

Demonstrate the use of various data types in a java program.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyses well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using data types in java.

D. Expected CourseOutcomes (COs):

CO-1: Understand OOP (Object-Oriented Programming) concepts with java.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Use various data types in a java program.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Data Types in Java

Data types specify the different sizes and values that can be stored in the variable. There are two

types of data types in Java:

- **Primitive data types:** The primitive data types include boolean, char, byte, short, int, long, float and double.
- **Non-primitive data types:** The non-primitive data types include Classes, Interfaces, and Arrays.

Java Primitive Data Types

In Java language, primitive data types are the building blocks of data manipulation. These are the most basic data types available in Java language.

- **Integers**
  This group includes byte, short, int, and long, which are for whole-valued signed numbers.
- **Floating-point numbers**
  This group includes float and double, which representnumbers with fractional precision.
- **Characters**
  This group includes char, which represents symbols in a character set,like letters and numbers.
- **Boolean**
  This group includes boolean, which is a special type for representingtrue/false values.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

…….

…….

…….

…….

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**Which is not an integer data type?** A. int B. long C. short D. double
Ans.	
Q. 2	**Write output for following code.** **class Main {** **public static void main(String args[]) {** **int t;** **System.out.println(t);** **}** **}** A. 0 B. Garbage value. C. Compile error. D. Runtime error.
Ans.	

N. References / Suggestions

1. https://www.javatpoint.com/java-data-types

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No:4	Develop a Java program to swap two numbers without using a temporary variable and with using a temporary variable (use command line argument to accept value from user).

A. Objective:

Swap two numbers with and without using a temporary variable and scan number using command line argument.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using command line argument.

D. Expected CourseOutcomes (COs):

CO-1: Understand OOP (Object-Oriented Programming) concepts with java.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Accept value using command line argument in a java program.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Java Command Line Arguments

- The java command-line argument is an argument i.e. passed at the time of running the java program.
- The arguments passed from the console can be received in the java program and it can be used as an input.
- So, it provides a convenient way to check the behavior of the program for the different values. You can pass N (1,2,3 and so on) numbers of arguments from the command prompt.

Simple example of command-line argument in java
In this example, we are receiving only one argument and printing it. To run this java program, you must pass at least one argument from the command prompt.

```
class CommandLineExample{
   public static void main(String args[]){
        System.out.println("Your first argument is: "+args[0]);
   }
}
```

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Nil

N. References / Suggestions

1. https://www.javatpoint.com/command-line-argument
2. https://www.geeksforgeeks.org/command-line-arguments-in-java/
3. https://www.programiz.com/java-programming/command-line-arguments

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No:5	Develop programs to demonstrate use of - 1) if statement and its different form 2) switch case statement

A. Objective:

Demonstrate the use of Decision-Making statements in java programs.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using Decision Making statements.

D. Expected CourseOutcomes (COs):

CO-1: Understand OOP (Object-Oriented Programming) concepts with java.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Use Decision Making statements in java programs.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Java If-else Statement

The Java if statement is used to test the condition. It checks boolean condition: true or false. There are various types of if statement in Java.

if statement

```
if (condition) {
   // block of code to be executed if the condition is true
}
```

if-else statement

```
if (condition) {
  // block of code to be executed if the condition is true
} else {
  // block of code to be executed if the condition is false
}
```

if-else-if ladder

```
if(condition1) {
   // block of code to be executed if condition1 is true
 } else if (condition2) {
   // block of code to be executed if the condition1 is false
   and condition2 is true
 } else {
   // block of code to be executed if the condition1 is false
   and condition2 is false
  }
```

nested if statement

```
if(condition){
  //code to be executed
       if(condition){
  //code to be executed
  }
 }
```

Java Switch Statement

The Java switch statement executes one statement from multiple conditions. It is like if-else-if ladder statement. The switch statement works with byte, short, int, long, enum types, String and some wrapper types like Byte, Short, Int, and Long. You can use strings in the switch statement.

Syntax:

```
switch(expression){
   case value1:
        //code to be executed;
   break;
        //optional
   case value2:
        //code to be executed;
   break;
        //optional
   ......
```

```
default:
     //code to be executed if all cases are not matched;
}
```

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Nil

N. References / Suggestions

1. https://www.youtube.com/watch?v=Tn6BNLD0PmU
2. https://www.javatpoint.com/java-if-else
3. https://www.javatpoint.com/java-switch

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No:6	Develop program to demonstrate use of- 1) for loop 2) 'while' and 'do while' loop

A. Objective:

Demonstrate the use of looping statements in java programs.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using looping statement.

D. Expected CourseOutcomes (COs):

CO-1: Understand OOP (Object-Oriented Programming) concepts with java.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Use looping statement in java programs.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Loop Statements

- In programming, sometimes we need to execute the block of code repeatedly while some condition evaluates to true. However, loop statements are used to execute the set of instructions in a repeated order. The execution of the set of instructions depends upon a particular condition.
- In Java, we have three types of loops that execute similarly. However, there are differences in their syntax and condition checking time.
  - for loop -while loop -do-while loop

Java for loop

In Java, for loop is similar to C and C++. It enables us to initialize the loop variable, check the condition, and increment/decrement in a single line of code. We use the for loop only when we exactly know the number of times, we want to execute the block of code.

```
for(initialization, condition, increment/decrement) {
   //block of statements
 }
```

Java while loop

- The while loop is also used to iterate over the number of statements multiple times. However, if we don't know the number of iterations in advance, it is recommended to use a while loop. Unlike for loop, the initialization and increment/decrement doesn't take place inside the loop statement in while loop.
- It is also known as the entry-controlled loop since the condition is checked at the start of the loop. If the condition is true, then the loop body will be executed; otherwise, the statements after the loop will be executed.
- The syntax of the while loop is given below.

```
while(condition){
   //looping statements
}
```

Java do-while loop

- The do-while loop checks the condition at the end of the loop after executing the loop statements. When the number of iterations is not known and we have to execute the loop at least once, we can use do-while loop.
- It is also known as the exit-controlled loop since the condition is not checked in advance. The syntax of the do-while loop is given below.

```
do {
   //statements
} while (condition);
```

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity

1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Nil

N. References / Suggestions

1. https://www.javatpoint.com/control-flow-in-java
2. https://www.youtube.com/watch?v=mzt5tmV7wxI

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No:7	Develop a Java program to find maximum and minimum numbers from array elements.

A. Objective:

Find maximum and minimum number from array elements.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using array in java.

D. Expected CourseOutcomes (COs):

CO-1: Understand OOP (Object-Oriented Programming) concepts with java.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Implement array to store element.
2. Find out maximum and minimum number from elements.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Java Arrays

- Normally, an array is a collection of similar type of elements which has contiguous memory location.
- Java array is an object which contains elements of a similar data type. Additionally, the elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.
- Array in Java is index-based, the first element of the array is stored at the 0^{th}index, 2nd element is stored on 1st index and so on.
- In Java, array is an object of a dynamically generated class. Java array inherits the Object class, and implements the Serializable as well as Cloneable interfaces. We can store primitive values or objects in an array in Java. Like C/C++, we can also create single dimensional or multidimensional arrays in Java.

Advantages

- Code Optimization: It makes the code optimized, we can retrieve or sort the data efficiently.
- Random access: We can get any data located at an index position.

Disadvantages

- **Size Limit:** We can store only the fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in Java which grows automatically.

Types of Array in java

- Single Dimensional Array
- Multidimensional Array

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

…….

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**Which of the following declares an array of integers named number?** A. int number; B. int [ ] number ; C. int new number [ ] ; D. int number = int [ ] ;
Ans.	
Q. 2	**What is the output of the following code fragment?** **int [ ] odd = {1, 3, 5, 7, 9, 11 };** **System.out.println(odd[0] + " " + odd[3]) ;** A. 1 5 B. 6 C. 1 7 D. 8
Ans.	
Q. 3	**Which of the following is FALSE about arrays on Java** A. A java array is always an object B. Length of array can be changed after creation of array C. Arrays in Java are always allocated on heap
Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=2j64wdFNWyE
2. https://www.javatpoint.com/array-in-java
3. https://www.geeksforgeeks.org/arrays-in-java/

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No:8	Develop a basic java program that demonstrates the use of Class & Object.

A. Objective:

Implement a java program to demonstrate the use of class and object.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using class and objects.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Implement class and object in a java program.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Objects and Classes in Java

- In object-oriented programming technique, we design a program using objects and classes.
- An object in Java is the physical as well as a logical entity, whereas, a class in Java is a logical entity only.

What is an object in Java?

- Object in Java:An entity that has state and behavior is known as an object e.g., chair, bike, marker, pen, table, car, etc. It can be physical or logical (tangible and intangible). The example of an intangible object is the banking system.
- An object is an instance of a class. A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.

Object Definitions:

- An object is a real-world entity.
- An object is a runtime entity.
- The object is an entity which has state and behavior.
- The object is an instance of a class.

What is a class in Java?

- A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.
- A class in Java can contain:
  - Fields
  - Methods
  - Constructors
  - Blocks
  - Nested class and interface

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Nil

N. References / Suggestions

1. https://www.youtube.com/watch?v=3sMxJxeUOLg
2. https://www.javatpoint.com/object-and-class-in-java

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No:9	Develop a java program to find the factorial of a given number using a recursive function.

A. Objective:

Find factorial of a given number using recursive function.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using recursion.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Implement recursive function.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Recursion in Java

Recursion in java is a process in which a method calls itself continuously. A method in java that calls itself is called recursive method.It makes the code compact but complex to understand.

Syntax:

```
returntypemethodname(){
   //code to be executed
   methodname();//calling same method
}
```

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Nil

N. References / Suggestions

1. https://www.javatpoint.com/recursion-in-java

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No:10	Develop a java program that demonstrates method overloading.

A. Objective:

Implement method overloading in a java program.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using method overloading.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Handle strings using different methods of String class.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Method Overloading in Java

- If a class has multiple methods having same name but different in parameters (different number of parameters, different types of parameters, or both), it is known as Method Overloading.
- If we have to perform only one operation, having same name of the methods increases the readability of the program.Method overloading is also known as Compile-time Polymorphism, Static Polymorphism, or Early binding in Java

Different Ways of Method Overloading in Java

- Changing the Number of Parameters.
- Changing Data Types of the Arguments.
- Changing the Order of the Parameters of Methods

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Nil

N. References / Suggestions

1. https://www.youtube.com/watch?v=KpwBVAYbPDA
2. https://www.javatpoint.com/method-overloading-in-java
3. https://www.geeksforgeeks.org/method-overloading-in-java/

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 11	Develop a program for implementation of different functions of String class.

A. Objective:

Use functions of String Class to handle strings in java.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using different methods of String class.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Handle strings using different methods of String class.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

String in Java

- Strings, which are widely used in Java programming, are a sequence of characters. In Java programming language, strings are treated as objects.
- The Java platform provides the String class to create and manipulate strings.
- The most direct way to create a string is to write –

```
String greeting = "Hello world!";
```

- Java String class provides a lot of methods to perform operations on strings such ascompare(), concat(), equals(), split(), length(), replace(), compareTo(), intern(), substring() etc.
- The java.lang.String class implements Serializable, Comparable and CharSequence interfaces.

How to create a string object?
There are two ways to create String object:
- By string literal
- By new keyword

String Literal
Java String literal is created by using double quotes. For Example:

```
String s="welcome";
```

- Each time you create a string literal, the JVM checks the "string constant pool" first. If the string already exists in the pool, a reference to the pooled instance is returned.
- If the string doesn't exist in the pool, a new string instance is created and placed in the pool. For example:

```
String s1="Welcome";
String s2="Welcome";//It doesn't create a new instance
```

By new keyword

```
String s=new String("Welcome");
//creates two objects and one reference variable
```

- In such case, JVM will create a new string object in normal (non-pool) heap memory, and the literal "Welcome" will be placed in the string constant pool. The variable s will refer to the object in a heap (non-pool).

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.

3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**What will be the output of below statements?** **String s = "Java String Quiz";** **System.out.println(s.charAt(s.toUpperCase().length()));** E. Convert “Z” to int 90 and prints “90”. F. Runtime Exception G. Prints “z”
Ans.	
Q. 2	**Select all valid methods of String class.** E. trim(). F. intern(). G. toLower().

	H. split().
Ans.	
Q. 3	**What will be the output of below statements?** **String s1 = "Cat";** **String s2 = "Cat";** **String s3 = new String("Cat");** **System.out.print(s1 == s2);** **System.out.print(s1 == s3);** A. truefalse. B. truetrue. C. falsefalse. D. falsetrue.
Ans.	

N. References / Suggestions

4. https://docs.oracle.com/javase/7/docs/api/java/lang/String.html
5. https://www.javatpoint.com/java-string
6. https://www.tutorialspoint.com/String-Handling-in-Java
7. https://www.youtube.com/watch?v=5J9o2Q7MqUQ&list=PLsyeobzWxl7rxNHxSwfErHYJG3en67TQG

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 12	Develop a program for implementation of Wrapper Class to convert primitive value into object(Boxing) and object into primitive value (Un-boxing).

A. Objective:

Use Wrapper class to convert primitive values into object and object into primitive type.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyse well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using wrapper class.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Convert primitive values into objects and objects into primitive values.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Wrapper classes in Java

- The wrapper class in Java provides the mechanism to convert primitive into object and object into primitive.
- Since J2SE 5.0, autoboxing and unboxing feature convert primitives into objects and objects into primitives automatically. The automatic conversion of primitive into an object is known as autoboxing and vice-versa unboxing.

Use of Wrapper classes in Java

Java is an object-oriented programming language, so we need to deal with objects many times like in Collections, Serialization, Synchronization, etc. Different scenarios, where you need to use the wrapper classes.

- **Change the value in Method:** Java supports only call by value. So, if we pass a primitive value, it will not change the original value. But, if we convert the primitive value in an object, it will change the original value.
- **Serialization:** We need to convert the objects into streams to perform the serialization. If we have a primitive value, we can convert it in objects through the wrapper classes.
- **Synchronization:** Java synchronization works with objects in Multithreading.
- **java.util package:** The java.util package provides the utility classes to deal with objects.

The eight classes of the java.lang package are known as wrapper classes in Java. The list of eight wrapper classes are given below:

Primitive Type	Wrapper class
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

Autoboxing

The automatic conversion of primitive data type into its corresponding wrapper class is known as autoboxing, for example, byte to Byte, char to Character, int to Integer, long to Long, float to Float, boolean to Boolean, double to Double, and short to Short.

Unboxing

The automatic conversion of wrapper type into its corresponding primitive type is known as Unboxing. It is the reverse process of autoboxing. Since Java 5, we do not need to use the intValue() method of wrapper classes to convert the wrapper type into primitives.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity

1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**Write the output:** **int arr[ ] = new int[5];** **System.out.println(arr[1]);** A. 1 B. 0 C. null D. Garbage Value
Ans.	
Q. 2	**List out wrapper classes provided by java.lang package.**

Ans.	

N. References / Suggestions

1. https://docs.oracle.com/javase/7/docs/api/java/sql/Wrapper.html
2. https://www.javatpoint.com/wrapper-class-in-java
3. https://www.geeksforgeeks.org/wrapper-classes-java/
4. https://www.w3schools.com/java/java_wrapper_classes.asp
5. https://www.youtube.com/watch?v=Fyc86kVIePE

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 13	Develop a program with a static block and show that it will be executed before the main( ) method in a class.

A. Objective:

Create a java program with static block and show that content of static block executed before main() method.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using package in java to solve given problems.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Implement static block in a java program.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Java static keyword

The static keyword in Java is used for memory management mainly. We can apply static keyword with variables, methods, blocks and nested classes. The static keyword belongs to the class than an instance of the class.

The static can be:

- Variable (also known as a class variable)
- Method (also known as a class method)
- Block and Nested class

Characteristics of static keyword:

- **Shared memory allocation:** Static variables and methods are allocated memory space only once during the execution of the program. This memory space is shared among all instances of the class, which makes static members useful for maintaining global state or shared functionality.
- **Accessible without object instantiation:** Static members can be accessed without the need to create an instance of the class. This makes them useful for providing utility functions and constants that can be used across the entire program.
- **Associated with class, not objects:** Static members are associated with the class, not with individual objects. This means that changes to a static member are reflected in all instances of the class, and that you can access static members using the class name rather than an object reference.
- **Cannot access non-static members:** Static methods and variables cannot access non-static members of a class, as they are not associated with any particular instance of the class.
- **Can be overloaded, but not overridden:** Static methods can be overloaded, which means that you can define multiple methods with the same name but different parameters. However, they cannot be overridden, as they are associated with the class rather than with a particular instance of the class.

Java static block Is used to initialize the static data member. It is executed before the main method at the time of classloading.

Example of static block

```
class A2{
     static{
          System.out.println("static block is invoked");
     }
          public static void main(String args[]){
          System.out.println("Hello main");
          }
}
```

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity

1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

NIL

N. References / Suggestions

1. https://www.youtube.com/watch?v=qRCEdWQ0f4Q
2. https://www.javatpoint.com/static-keyword-in-java
3. https://www.geeksforgeeks.org/static-keyword-java/
4. https://www.w3schools.com/java/ref_keyword_static.asp

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 14	Develop a program to demonstrate use of static functions.

A. Objective:

Create a java program with static method and demonstrate its use.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using static function to use function without creating object of the class.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Create and access static method in a java class.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Java static method

When a method is declared with the static keyword, it is known as the static method. The most common example of a static method is the main( ) method. As discussed above, any static member can be accessed before any objects of its class are created, and without reference to any object. Methods declared as static have several restrictions:

- They can only directly call other static methods.
- They can only directly access static data.
- They cannot refer to this or super in any way.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

…….

…….

…….

…….

…….

…….

…….

…….

…….

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Nil

N. References / Suggestions

1. https://www.youtube.com/watch?v=qRCEdWQ0f4Q
2. https://www.javatpoint.com/static-keyword-in-java
3. https://www.geeksforgeeks.org/static-keyword-java/
4. https://www.w3schools.com/java/ref_keyword_static.asp

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 15	Develop a program to demonstrate use of 'this' keyword. Check whether 'this' can access the private members of the class or not.

A. Objective:

Use 'this' keyword to access member variable of class with different access modifiers.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving using 'this' keyword in java.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Use 'this' keyword in java program to avoid shadow problem.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

this keyword in Java

There can be a lot of usage of Java this keyword. In Java, this is a reference variable that refers to the current object.

Usage of this keyword in Java

Here is given the 6 usage of java this keyword.

- this can be used to refer current class instance variable.
- this can be used to invoke current class method (implicitly)
- this() can be used to invoke current class constructor.
- this can be passed as an argument in the method call.
- this can be passed as argument in the constructor call.
- this can be used to return the current class instance from the method.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**The keyword this in Java represents the current _____ you are executing within.** A. Class B. Method C. Program D. Object
Ans.	
Q. 2	**Which of the following operators is used with the keyword this?** A. -> B. . C. + D. ^
Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=k_SJOnVa_G0
2. https://www.javatpoint.com/this-keyword
3. https://www.w3schools.com/java/ref_keyword_this.asp
4. https://www.guru99.com/java-this-keyword.html

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 16	**Develop a program with an overloaded constructor. Also develop the copy constructor to create a new object with the state of the existing object.**

A. Objective:

Demonstrate the use of overloaded constructors and perform object cloning using copy constructor.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving using constructor in java.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Implement constructor to initialize the objects.
2. Overload the constructors.
3. Clone objects.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Constructors in Java

- In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling constructor, memory for the object is allocated in the memory.
- It is a special type of method which is used to initialize the object.
- Every time an object is created using the new() keyword, at least one constructor is called.
- It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default.

Rules for creating Java constructor

There are three rules defined for the constructor.

- Constructor name must be the same as its class name
- A Constructor must have no explicit return type
- A Java constructor cannot be abstract, static, final, and synchronized

There are two types of constructors in Java:

- Default constructor (no-arg constructor)
- Parameterized constructor

Java Default Constructor

- A constructor is called "Default Constructor" when it doesn't have any parameter.
- Syntax of default constructor:

```
<class_name>(){}
```

Java Parameterized Constructor

A constructor which has a specific number of parameters is called a parameterizedconstructor.

Why use the parameterized constructor?

The parameterized constructor is used to provide different values to distinct objects. However, you can provide the same values also.

Constructor Overloading in Java

- In Java, a constructor is just like a method but without return type. It can also be overloaded like Java methods.
- Constructor overloading in Java is a technique of having more than one constructor with different parameter lists. They are arranged in a way that each constructor performs a different task. They are differentiated by the compiler by the number of parameters in the list and their types.

Java Copy Constructor

- There is no copy constructor in Java. However, we can copy the values from one object to another like copy constructor in C++.
- There are many ways to copy the values of one object into another in Java. They are:
  - By constructor
  - By assigning the values of one object into another
  - By clone() method of Object class

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**Which of the following is/are true about constructors in Java?** **I. Constructor name should be same as class name.** **II. If you don't define a constructor for a class, a default parameterless constructor is automaticallycreated by the compiler.** **III. The default constructor calls super() and initializes all instance variables to default value like 0, null.** **IV. If we want to parent class constructor, it must be called in first line of constructor.** A. I B. I,II C. I,II,III D. I,II,III,VI
Ans.	
Q. 2	**How many arguments in a default constructor?** A. 0 B. 1 C. 2 D. 3
Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=UC_aqNUFyVI
2. https://www.javatpoint.com/java-constructor
3. https://www.w3schools.com/java/java_constructors.asp
4. https://www.geeksforgeeks.org/constructors-in-java/

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 17	**Develop a program to demonstrate the use of private constructor and also write a method which will count the number of instances created using default constructor only.**

A. Objective:

Demonstrate the use of private constructor.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using private constructor.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Implement Private constructor in a java program.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Private Constructor in Java

- In Java, the constructor is a special type of method that has the same name as the class name. Internally, a constructor is always called when we create an object of the class. It is used to initialize the state of an object.
- In the same way, Java also allows us to create a private constructor. In this section, we will

discuss private constructors in Java, rules for creating a private constructor, and its use cases. Also, we will see its implementation.

What is a private constructor?

Java allows us to declare a constructor as private. We can declare a constructor private by using the private access specifier. Note that if a constructor is declared private, we are not able to create an object of the class. Instead, we can use this private constructor in Singleton Design Pattern.

Rules for Private Constructor

The following rules keep in mind while dealing with private constructors.

- It does not allow a class to be sub-classed.
- It does not allow to create an object outside the class.
- If a class has a private constructor and when we try to extend the class, a compile-time error occurs.
- We cannot access a private constructor from any other class.
- If all the constant methods are there in our class, we can use a private constructor.
- If all the methods are static then we can use a private constructor.
- We can use a public function to call the private constructor if an object is not initialized.
- We can return only the instance of that object if an object is already initialized.

Use Cases of Private Constructor

The main purpose of using a private constructor is to restrict object creation. We also use private constructors to implement the singleton design pattern. The use-cases of the private constructor are as follows:

- It can be used with static members-only classes.
- It can be used with static utilityor constant classes.
- It can also be used to create singleton classes.
- It can be used to assign a name, for instance, creation by utilizing factory methods.
- It is also used to avoid sub-classing.
- It incorporates the factory methods.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.

3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

N.A.

N. References / Suggestions

1. https://www.youtube.com/watch?v=UC_aqNUFyVI
2. https://www.javatpoint.com/private-constructor-in-java
3. https://www.baeldung.com/java-private-constructors

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 18	Develop a program to demonstrate single inheritance, multilevel inheritance, and hierarchical inheritance.

A. Objective:

Demonstrate the use of various types of inheritance in a java program.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solvingskills by implementing inheritance.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Extend a class.
2. Implement different types of inheritance.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Inheritance in Java

- In java programming, multiple and hybrid inheritance is supported through interface Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).
- The idea behind inheritance in Java is that you can create new classes that are built upon

existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

- Inheritance represents the IS-A relationship which is also known as a parent-child relationship.

Why use inheritance in java

- For Method Overriding (so runtime polymorphism can be achieved).
- For Code Reusability.

Terms used in Inheritance

- Class: A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.
- Sub Class/Child Class: Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
- Super Class/Parent Class: Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
- Reusability: As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class. You can use the same fields and methods already defined in the previous class.

The syntax of Java Inheritance

```
class Subclass-name extends Superclass-name
{
   //methods and fields
}
```

The extends keyword indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.

Types of inheritance in java

- On the basis of class, there can be three types of inheritance in java: single, multilevel and hierarchical.
- In java programming, multiple and hybrid inheritance is supported through interface only.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.

2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**What type of inheritance does Java have?** A. Single Inheritance B. Double Inheritance C. Multiple Inheritance D. Class Inheritance
Ans.	
Q. 2	**Which of these keyword must be used to inherit a class?** A. extent B. super C. this D. extends

Ans.	
Q. 3	**Which of these class is related to all the exceptions that are explicitly Which of the following is true about inheritance in Java?** **1) Private methods are final.** **2) Protected members are accessible within a package and inherited classes outside the package.** **3) Protected methods are final.** **4) We cannot override private methods.** A. 1, 2 and 4 B. Only 1 and 2 C. 1, 2 and 3 D. 2, 3 and 4
Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=rxsl1TzcEgg&list=PLbRMhDVUMngcx5xHChJ-f7ofxZI4JzuQR&index=14
2. https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html
3. https://www.javatpoint.com/inheritance-in-java
4. https://www.geeksforgeeks.org/inheritance-in-java/
5. https://www.tutorialspoint.com/java/java_inheritance.htm

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 19	**Develop a program with one class named shape which has two member functions named erase () and draw(). In the program we have three other subclasses: circle, triangle and square. Override methods of the super class into sub classes.**

A. Objective:

Implement method overriding in a java program to override method of super class into sub class.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solvingskills using method overriding.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Override a method of super class into sub class.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Method Overriding in Java

- If subclass (child class) has the same method as declared in the parent class, it is known as method overriding in Java.
- In other words, if a subclass provides the specific implementation of the method that has been

declared by one of its parent class, it is known as method overriding.

Usage of Java Method Overriding

- Method overriding is used to provide the specific implementation of a method which is already provided by its superclass.
- Method overriding is used for runtime polymorphism

Rules for Java Method Overriding

- The method must have the same name as in the parent class
- The method must have the same parameter as in the parent class.
- There must be an IS-A relationship (inheritance).

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

N.A.

N. References / Suggestions

1. https://www.youtube.com/watch?v=CLzgS08equQ
2. https://www.javatpoint.com/method-overriding-in-java
3. https://www.geeksforgeeks.org/overriding-in-java/

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 20	Develop a program for implementation of Dynamic method dispatch.

A. Objective:

Implement runtime polymorphism in java program.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solvingskills using runtime polymorphism.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Implement runtime polymorphism in a java program.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

- Polymorphism in Java is a concept by which we can perform a single action in different ways. Polymorphism is derived from 2 Greek words: poly and morphs. The word "poly" means many and "morphs" means forms. So polymorphism means many forms.
- There are two types of polymorphism in Java: compile-time polymorphism and runtime

polymorphism. We can perform polymorphism in java by method overloading and method overriding.

- If you overload a static method in Java, it is the example of compile time polymorphism. Here, we will focus on runtime polymorphism in java.

Runtime Polymorphism in Java

- Runtime polymorphism or Dynamic Method Dispatch is a process in which a call to an overridden method is resolved at runtime rather than compile-time.
- Threads are independent. If there occurs exception in one thread, it doesn't affect other threads. It uses a shared memory area.
- In this process, an overridden method is called through the reference variable of a superclass. The determination of the method to be called is based on the object being referred to by the reference variable.
- Let's first understand the upcasting before Runtime Polymorphism.

Upcasting

- If the reference variable of Parent class refers to the object of Child class, it is known as upcasting.
- For example:

```
class A{}
class B extends A{}
A a=new B();//upcasting
```

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

…….

…….

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

NIL

N. References / Suggestions

1. https://www.youtube.com/watch?v=eIO9dNLkhDQ
2. https://www.javatpoint.com/runtime-polymorphism-in-java
3. https://www.geeksforgeeks.org/dynamic-method-dispatch-runtime-polymorphism-java/

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 21	Develop a java program that demonstrates the use of Abstract class.

A. Objective:

Apply the concept of abstraction to solve particular problem.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using abstract class (to provide data abstraction) in java to solve given problems.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Develop programs using abstract classes to provide data abstraction.

F. Expected Affective Domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Abstraction in Java

Abstraction is a process of hiding the implementation details and showing only functionality to the user. Another way, it shows only essential things to the user and hides the internal details, for example, sending SMS where you type the text and send the message. You don't know the internal

processing about the message delivery.Abstraction lets you focus on what the object does instead of how it does it.

Abstract class in Java

- A class which is declared with the abstract keyword is known as an abstract class in Java.
- It can have abstract and non-abstract methods (method with the body).

Ways to achieve Abstraction

There are two ways to achieve abstraction in java

- Abstract class (0 to 100%)
- Interface (100%)

Abstract class:

- A class which is declared as abstract is known as an abstract class. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be instantiated.
- Points to Remember
  - An abstract class must be declared with an abstract keyword.
  - It can have abstract and non-abstract methods.
  - It cannot be instantiated.
  - It can have constructors and static methods also.
  - It can have final methods which will force the subclass not to change the body of the method.

Example of abstract class

```
abstract class A
{
        ……… .
        ……… .
}
```

Abstract Method in Java

A method which is declared as abstract and does not have implementation is known as an abstract method.

Example of abstract method

```
abstract void printStatus();//no method body and abstract
```

Example of Abstract class that has an abstract method

In this example, Bike is an abstract class that contains only one abstract method run.Its implementation is provided by the Honda class.

```
abstract class Bike
{
      abstract void run();
}

class Honda extends Bike
{
```

```
        void run()
        {
            System.out.println("running safely");
        }
        public static void main(String args[])
        {
            Bike obj = new Honda();
            obj.run();
        }
    }
```

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**What is an abstract class?** A. An abstract class is one without any child classes. B. An abstract class is any parent class with more than one child class. C. An abstract class is a class which has at least one abstract method which cannot be instantiated. D. Is a "base class".
Ans.	
Q. 2	**Can an abstract class define both abstract methods and non-abstract methods?** A. Yes - the child classes will inherit both. B. Yes - but the child classes do not inherit the abstract methods. C. No--it must have all abstract methods. D. No--it must have all one or the other.
Ans.	
Q. 3	**Can an abstract parent class have non-abstract children?** A. No—an abstract parent must have only abstract children.. B. No—an abstract parent must have no children at all. C. Yes—all children of an abstract parent must be non-abstract. D. Yes—an abstract parent can have both abstract and non-abstract children.
Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=qgBv1_Plldo&list=PLfn3cNtmZdPOe3R_wO_h540QNfMkCQ0ho&index=20
2. https://www.youtube.com/watch?v=p_4Dyfplqkw
3. https://www.youtube.com/watch?v=RcIsb9iFKH8
4. https://docs.oracle.com/javase/tutorial/java/IandI/abstract.html
5. https://www.javatpoint.com/abstract-class-in-java

6. https://www.w3schools.com/java/java_abstract.asp

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 22	**Develop a java program that illustrates interface inheritance. Interface 'A1' and 'A2' are extended from interface 'A'. Interface 'A12' inherited from both 'A1' and 'A2'. Each interface declares one method and one constant. Class 'Interface_Imple' implements 'A12'. Instantiate 'Interface_Imple' and invoke each of its methods. Each method displays one of the constants.**

A. Objective:

Apply the concept of interface to provide full abstraction.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using interface (to provide data abstraction) in java to solve given problems.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Develop programs using interface to provide data abstraction.
2. Extend one interface from another interface.
3. Implement concrete class by overriding all methods of interfaces.

F. Expected Affective domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Interface in Java

- An interface in Java is a blueprint of a class. It has static constants and abstract methods.
- The interface in Java is a mechanism to achieve abstraction. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java.
- In other words, you can say that interfaces can have abstract methods and variables. It cannot have a method body.
- Java Interface also represents the IS-A relationship.

Why use Java interface?

- There are mainly three reasons to use interface. They are given below.
- It is used to achieve abstraction.
- By interface, we can support the functionality of multiple inheritance.
- It can be used to achieve loose coupling.

How to declare an interface?

An interface is declared by using the interface keyword. It provides total abstraction; means all the methods in an interface are declared with the empty body, and all the fields are public, static and final by default. A class that implements an interface must implement all the methods declared in the interface.

Syntax:

```
interface <interface_name>
{
     // declare constant fields
     // declare methods that abstract
     // by default.
}
```

The relationship between classes and interfaces

As shown in the figure given below, a class extends another class, an interface extends another interface, but a class implements an interface.

Java Interface Example

In this example, the Printable interface has only one method, and its implementation is provided in the A class.

```
interface printable{
     public void print();
}

class A implements printable{

     public void print(){
          System.out.println("Hello");
     }

     public static void main(String args[]){
          A obj = new A();
          obj.print();
}
}
```

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**Which of these keywords is used to define interfaces in Java?** A. interface B. Interface C. intf D. Intf
Ans.	
Q. 2	**Which of these access specifiers can be used for an interface?** A. Public B. Protected C. private D. All of the mentioned
Ans.	
Q. 3	**A Java Class inherits Constants and Methods of an Interface using ____ keyword.** A. INTERFACE B. IMPLEMENTS C. EXTENDS D. All the above
Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=qgBv1_Plldo&list=PLfn3cNtmZdPOe3R_wO_h540QNfMkCQ0ho&index=20
2. https://www.youtube.com/watch?v=A1uqgEz3hB0
3. https://www.youtube.com/watch?v=AG_7wWFBquQ
4. https://docs.oracle.com/javase/tutorial/java/IandI/createinterface.html
5. https://www.javatpoint.com/interface-in-java

6. https://www.w3schools.com/java/java_interface.asp

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 23	Develop a program to create a Package and demonstrate how packages are used in java. And use java access modifier to demonstrate the access rules in a package.

A. Objective:

Create java packages, arrange class files into packages, and use access modifier to demonstrate access rules in packages.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using package in java to solve given problems.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Develop programs to create package.
2. Use java access modifier.
3. Demonstrate the access rules in package.

F. Expected Affective domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Java Package

- A java package is a group of similar types of classes, interfaces and sub-packages.
- Package in java can be categorized in two form, built-in package and user-defined package.
- There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Advantage of Java Package

- Java package is used to categorize the classes and interfaces so that they can be easily maintained.
- Java package provides access protection.
- Java package removes naming collision.

Simple example of java package

The package keyword is used to create a package in java.

save as Simple.java

```
package mypack;

public class Simple{
     public static void main(String args[]){
          System.out.println("Welcome to package");
     }
}
```

How to compile java package

If you are not using any IDE, you need to follow the syntax given below:

javac -d directory javafilename

For example

javac -d . Simple.java

The -d switch specifies the destination where to put the generated class file. You can use any directory name like /home (in case of Linux), d:/abc (in case of windows) etc. If you want to keep the package within the same directory, you can use . (dot).

Access package from another package?

There are three ways to access the package from outside the package.

- import package.*;
- import package.classname;
- fully qualified name.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA

2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

K. Practical related Quiz/Questions.

Q. 1	**Which of this access specifies can be used for a class so that its members can be accessed by a different class in the same package?** A. Public B. Protected C. No Modifier D. All of the mentioned
Ans.	
Q. 2	**Which of the following is an incorrect statement about packages?** A. Package defines a namespace in which classes are stored B. A package can contain other package within it C. Java uses file system directories to store packages D. A package can be renamed without renaming the directory in which the classes are stored
Ans.	
Q. 3	**Which of these keywords is used to define packages in Java?** A. Pkg B. pkg C. Package D. Package

Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=TwU3cv1FFis&list=PLbRMhDVUMngcx5xHChJ-f7ofxZI4JzuQR&index=18
2. https://www.youtube.com/watch?v=Bua6LQO2vQ8
3. https://docs.oracle.com/javase/8/docs/api/java/lang/Package.html
4. https://www.javatpoint.com/package

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 24	Develop a program to demonstrate the use of 'super' and 'final' keywords.

A. Objective:

Use "super" keyword to access member of parent class and make variable, method and class as final using "final" keyword.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills using "super" and "final" keyword in java.

D. Expected CourseOutcomes (COs):

CO-2: Understand building blocks of OOPs language, inheritance, package and interfaces.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Access parent class's member using "super" keyword.
2. Apply restriction on member variable, member method and class using "final" keyword.

F. Expected Affective domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Super Keyword in Java

- The super keyword in Java is a reference variable which is used to refer immediate parent class object.

- Whenever you create the instance of subclass, an instance of parent class is created implicitly which is referred by super reference variable.

Usage of Java super Keyword
- super can be used to refer immediate parent class instance variable.
- super can be used to invoke immediate parent class method.
- super() can be used to invoke immediate parent class constructor.

Final Keyword In Java
The final keyword in java is used to restrict the user. The java final keyword can be used in many context.

Final can be:
- variable
- method
- class

The final keyword can be applied with the variables, a final variable that have no value it is called blank final variable or uninitialized final variable. It can be initialized in the constructor only. The blank final variable can be static also which will be initialized in the static block only. We will have detailed learning of these. Let's first learn the basics of final keyword.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

…….

…….

…….

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**What is the use of final keyword in Java?** A. When a class is made final, a subclass of it can not be created. B. When a method is final, it can not be overridden. C. When a variable is final, it can be assigned value only once. D. All of the above
Ans.	
Q. 2	**Define "final" keyword.**
Ans.	
Q. 3	**Can we call parent class constructor in sub class constructor i.e constructor chaining by using super keyword?**
Ans.	

N. References / Suggestions

5. https://www.youtube.com/watch?v=1ZDApgnufSI
6. https://www.youtube.com/watch?v=IPJYL18NCpw
7. https://www.youtube.com/watch?v=WZgcRSWVgpQ
8. https://www.youtube.com/watch?v=OBYeDzgsOxc
9. https://docs.oracle.com/javase/tutorial/java/IandI/final.html
10. https://www.javatpoint.com/super-keyword
11. https://www.javatpoint.com/final-keyword

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 25	Develop programs to demonstrate the use of Exception Handling using predefined Exception Classes.

A. Objective:

Use inbuilt exception classes to handle Arithmetic and ArrayIndexOutofBoundexception.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving and debugging skills by applying exception handling in java.

D. Expected CourseOutcomes (COs):

CO-3: Implement exception handling and multithreading in object-oriented programs.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Handle exception in java program using predefined exception classes.

F. Expected Affective domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Exception Handling in Java

- The Exception Handling in Java is one of the powerful mechanisms to handle the runtime errors so that the normal flow of the application can be maintained.
- In this practical, we will learn about Java exceptions, it's types, and the difference between checked and unchecked exceptions.

What is Exception in Java?
Exception is an abnormal condition.In Java, an exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.
What is Exception Handling?
Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc.

Advantage of Exception Handling
The core advantage of exception handling is to maintain the normal flow of the application. An exception normally disrupts the normal flow of the application; that is why we need to handle exceptions.

Types of Java Exceptions
There are mainly two types of exceptions: checked and unchecked. An error is considered as the unchecked exception. There are three basic types of exceptions namely:

- Checked Exception
- Unchecked Exception
- Error

Difference between Checked and Unchecked Exceptions

Checked Exception
The classes that directly inherit the Throwable class except RuntimeException and Error are known as checked exceptions. For example, IOException, SQLException, etc. Checked exceptions are checked at compile-time.

Unchecked Exception
The classes that inherit the RuntimeException are known as unchecked exceptions. For example, Arithmetic Exception, NullPointerException, ArrayIndexOutOfBounds Exception, etc. Unchecked exceptions are not checked at compile-time, but they are checked at runtime.

Error
Error is irrecoverable. Some example of errors is OutOfMemoryError, VirtualMachineError, AssertionError etc.

Java Exception Keywords
Java provides five keywords that are used to handle the exception. The following table describes each.

Keyword	Description
try	The "try" keyword is used to specify a block where we should place an exception code. It means we can't use try block alone. The try block must be followed by either catch or finally.
catch	The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.
finally	The "finally" block is used to execute the necessary code of the program. It is executed whether an exception is handled or not.
throw	The "throw" keyword is used to throw an exception.

throws	The "throws" keyword is used to declare exceptions. It specifies that there may occur an exception in the method. It doesn't throw an exception. It is always used with method signature.

Java Exception Handling Example

```
public class JavaExceptionExample{
     public static void main(String args[]){
          try{
               //code that may raise exception
               int data=100/0;
     }catch(ArithmeticException e){
               System.out.println(e);
          }
          //rest code of the program
          System.out.println("rest of the code...");
          }
     }
}
```

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**Which of the following are checked exceptions?** E. Exception F. IllegalArgumentException G. IOException H. NullPointerException I. NumberFormatException J. StackOverflowError
Ans.	
Q. 2	**Which of the following exceptions are thrown by the JVM?** A. ArrayIndexOutOfBoundsException B. ExceptionInInitializerError C. java.io.IOException D. NullPointerException E. NumberFormatException
Ans.	
Q. 3	**Which of these is a super class of all errors and exceptions in the Java language?** A. RunTimeExceptions B. Throwable C. Catchable D. None of the above
Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=vUov8EkjZjU&list=PLbRMhDVUMngcx5xHChJ-f7ofxZI4JzuQR&index=24

2. https://www.youtube.com/watch?v=W-N2ltgU-X4
3. https://www.youtube.com/watch?v=9kI0_F41i5Y
4. https://www.javatpoint.com/exception-handling-in-java
5. https://docs.oracle.com/javase/tutorial/essential/exceptions/index.html

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 26	Develop a program to handle multiple exceptions using multiple try blocks and multiple catch blocks.

A. Objective:

Use exception handling keywords to handle multiple exceptions in a single java program.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving and debugging skills by applying exception handling in java.

D. Expected CourseOutcomes (COs):

CO-3: Implement exception handling and multithreading in object-oriented programs.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Handle multiple exceptions in java program using predefined exception classes.

F. Expected Affective domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Java Catch Multiple Exceptions

Java Multi-catch block

A try block can be followed by one or more catch blocks. Each catch block must contain a different exception handler. So, if you have to perform different tasks at the occurrence of different exceptions, use java multi-catch block.

Points to remember

- At a time only one exception occurs and at a time only one catch block is executed.

- All catch blocks must be ordered from most specific to most general, i.e. catch for ArithmeticException must come before catch for Exception.

Flowchart of Multi-catch Block

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

N.A.

N. References / Suggestions

1. https://www.youtube.com/watch?v=vUov8EkjZjU&list=PLbRMhDVUMngcx5xHChJ-f7ofxZI4JzuQR&index=24
2. https://www.javatpoint.com/multiple-catch-block-in-java
3. https://docs.oracle.com/javase/tutorial/essential/exceptions/index.html

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 27	Develop a program to implement user defined exceptions.

A. Objective:

Create user defined exception class to handle user defined exception.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving and debugging skills by applying exception handling in java.

D. Expected CourseOutcomes (COs):

CO-3: Implement exception handling and multithreading in object-oriented programs.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Handle user defined exception in java program exception classes.

F. Expected Affective domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

User defined Exception Handling in Java

- In Java, we can create our own exceptions that are derived classes of the Exception class. Creating our own Exception is known as custom exception or user-defined exception. Basically, Java custom exceptions are used to customize the exception according to user need.
- Consider the example 1 in which InvalidAgeException class extends the Exception class.
- Using the custom exception, we can have your own exception and message. Here, we have passed a string to the constructor of superclass i.e. Exception class that can be obtained using getMessage() method on the object we have created.

Why use custom exceptions?

Java exceptions cover almost all the general type of exceptions that may occur in the programming. However, we sometimes need to create custom exceptions.

Following are few of the reasons to use custom exceptions:

- To catch and provide specific treatment to a subset of existing Java exceptions.
- Business logic exceptions: These are the exceptions related to business logic and workflow. It is useful for the application users or the developers to understand the exact problem.
- In order to create custom exception, we need to extend Exception class that belongs to java.lang package.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

N.A.

N. References / Suggestions

1. https://www.youtube.com/watch?v=vUov8EkjZjU&list=PLbRMhDVUMngcx5xHChJ-f7ofxZI4JzuQR&index=24
2. https://www.javatpoint.com/custom-exception
3. https://docs.oracle.com/javase/tutorial/essential/exceptions/index.html

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 28	Develop a program to demonstrate use of throw, throws, and finally keyword.

A. Objective:

Use throw, throws and finally keyword in java program to throw exception explicitly and to execute some clean up code in finally block.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving and debugging skills by applying exception handling in java.

D. Expected CourseOutcomes (COs):

CO-3: Implement exception handling and multithreading in object-oriented programs.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Use throw keyword to throw an exception explicitly.
2. Use finally and throws keyword.

F. Expected Affective domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

- Throw, throws and finally are the keywords in Java that are used in exception handling. The throw keyword is used to throw an exception and throws is used to declare the list of possible exceptions with the method signature.
- Whereas finally block is used to execute essential code, specially to release the occupied resources.

Java Throw

The throw keyword is used to throw an exception explicitly. Only object of Throwable class or its sub classes can be thrown.

Program execution stops on encountering throw statement, and the closest catch statement is checked for matching type of exception.

Syntax:

```
throw ThrowableInstance
```

example

```
throw NullPointerException();
```

Java throws Keyword

- The throws keyword is used to declare the list of exception that a method may throw during execution of program.
- Any method that is capable of causing exceptions must list all the exceptions possible during its execution, so that anyone calling that method gets a prior knowledge about which exceptions are to be handled.
- A method can do so by using the throws keyword.

Syntax:

```
type method_name(parameter_list) throws exception_list
{
  // definition of method
}
```

finally clause

- A finally keyword is used to create a block of code that follows a try block. A finally block of code is always executed whether an exception has occurred or not.
- Using a finally block, it lets you run any cleanup type statements that you want to execute, no matter what happens in the protected code.
- A finally block appears at the end of catch block.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**Which of these keywords is used to generate an exception explicitly?** E. finally F. throw G. catch H. try
Ans.	
Q. 2	**Which of these operator is used to generate an instance of an exception than can be thrown by using throw?** E. new F. malloc G. alloc H. thrown
Ans.	
Q. 3	**Which of these class is related to all the exceptions that are explicitly thrown?** A. Error B. Exception C. Throwable D. Throw
Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=W4Zve_x0o30&list=PLbRMhDVUMngcx5xHChJ-f7ofxZI4JzuQR&index=26
2. https://www.studytonight.com/java/throw-throws-and-finally-keyword.php
3. https://www.geeksforgeeks.org/try-catch-throw-and-throws-in-java/
4. https://www.javatpoint.com/difference-between-throw-and-throws-in-java
5. https://docs.oracle.com/javase/tutorial/essential/exceptions/index.html

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 29	**Develop a program that executes two threads. One thread displays "Java Programming" every 2 seconds, and the other displays "Semester - 4th" every 5 seconds.(Create the threads by extending the Thread class).**

A. Objective:

Implement multithreading in a java program using Thread class.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solvingskills using Thread class to implement multithreading in a java program.

D. Expected CourseOutcomes (COs):

CO-3: Implement exception handling and multithreading in object-oriented programs.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Create thread in java program by extending Thread class.

F. Expected Affective domain Outcome(ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Thread in java

- A thread is a lightweight subprocess, the smallest unit of processing. It is a separate path of execution.
- Threads are independent. If there occurs exception in one thread, it doesn't affect other threads. It uses a shared memory area.

Java Thread class

- Java provides Thread class to achieve thread programming. Thread class provides constructors and methods to create and perform operations on a thread.
- Thread class extends Object class and implements Runnable interface.

Life cycle of a Thread (Thread States)

In Java, a thread always exists in any one of the following states. These states are:

- New
- Active
- Blocked / Waiting
- Timed Waiting
- Terminated

Ways to create a thread in Java

There are two ways to create a thread:

- By extending Thread class
- By implementing Runnable interface.

Thread class:

- Thread class provide constructors and methods to create and perform operations on a thread.Thread class extends Object class and implements Runnable interface.
- Commonly used Constructors of Thread class:
  - Thread()
  - Thread(String name)
  - Thread(Runnable r)
  - Thread(Runnable r,String name)

Java Thread Example by extending Thread class

```
class Multi extends Thread{
     public void run(){
          System.out.println("thread is running...");
     }

     public static void main(String args[]){
          Multi t1=new Multi();
          t1.start();
     }
}
```

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**How many ways a thread can be created in Java multithreading?** A. 1 B. 2 C. 5 D. 10
Ans.	

Q. 2	**What requires less resources?** A. Thread B. Process C. Neither Thread nor Process
Ans.	
Q. 3	**Which of the following is a correct constructor for thread?** A. Thread(Runnable a, String str) B. Thread(int priority) C. Thread(Runnable a, int priority) D. Thread(Runnable a, ThreadGroup t)
Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=6rYOyIGfy3w&list=PLbRMhDVUMngcx5xHChJ-f7ofxZI4JzuQR&index=28
2. https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.html
3. https://www.geeksforgeeks.org/java-threads/
4. https://www.javatpoint.com/how-to-create-a-thread-in-java

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 30	**Develop a program that executes two threads. One thread will print the even numbers and the another thread will print odd numbers between 1 to 10.(Create the thread by implementing Runnable interface)**

A. Objective:

Implement multithreading in a java program using Runnable interface.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solvingskills using Runnable interface to implement multithreading in a java program.

D. Expected CourseOutcomes (COs):

CO-3: Implement exception handling and multithreading in object-oriented programs.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Create thread in java program by extending Runnable interface.

F. Expected Affective domain Outcome(ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Thread in java

- A thread is a lightweight subprocess, the smallest unit of processing. It is a separate path of execution.
- Threads are independent. If there occurs exception in one thread, it doesn't affect other threads. It uses a shared memory area.

Java Thread class

- Java provides Thread class to achieve thread programming. Thread class provides constructors and methods to create and perform operations on a thread.
- Thread class extends Object class and implements Runnable interface.

Life cycle of a Thread (Thread States)

In Java, a thread always exists in any one of the following states. These states are:

- New
- Active
- Blocked / Waiting
- Timed Waiting
- Terminated

Ways to create a thread in Java

There are two ways to create a thread:

- By extending Thread class
- By implementing Runnable interface.

Java Runnable Interface

- Java runnable is an interface used to execute code on a concurrent thread. It is an interface which is implemented by any class if we want that the instances of that class should be executed by a thread.
- The runnable interface has an undefined method run() with void as return type, and it takes in no arguments.

How to create a thread using Runnable interface

To create a thread using runnable, use the following code-

```
Runnable runnable = new MyRunnable();
Thread thread = new Thread(runnable);
thread.start();
```

A simple thread example using runnable

```
public class ExampleClass implements Runnable {

     @Override
     public void run() {
          System.out.println("Thread has ended");
     }

     public static void main(String[] args) {
          ExampleClass ex = new ExampleClass();
     Thread t1= new Thread(ex);
          t1.start();
     System.out.println("Hi");
     }
}
```

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**Which of these method is used to implement Runnable interface?** A. stop() B. run() C. runThread() D. stopThread()
Ans.	

Q. 2	**Which of these method is used to begin the execution of a thread?** A. run() B. start() C. runThread() D. startThread()
Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=Kni0HMvq1Ig&list=PLbRMhDVUMngcx5xHChJ-f7ofxZI4JzuQR&index=29
2. https://www.geeksforgeeks.org/runnable-interface-in-java/

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No: 31	Develop a program to demonstrate use of synchronization of threads when multiple threads are trying to update a common variable.

A. Objective:

Create more than one thread in a java program and provide synchronization between them.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills by providing synchronization between multiple threads in a java program.

D. Expected CourseOutcomes (COs):

CO-3: Implement exception handling and multithreading in object-oriented programs.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Create multiple threads in a java program.
2. Provide synchronization between multiple threads.

F. Expected Affective domain Outcome (ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

Synchronization in Java

- Synchronization in Java is the capability to control the access of multiple threads to any shared resource.
- Java Synchronization is better option where we want to allow only one thread to access the shared resource.

Thread Synchronization

- There are two types of thread synchronization mutual exclusive and inter-thread communication.
- Mutual Exclusive
  - Synchronized method.
  - Synchronized block.
  - Static synchronization.
- Cooperation (Inter-thread communication in java)

Mutual Exclusive

- Mutual Exclusive helps keep threads from interfering with one another while sharing data. It can be achieved by using the following three ways:
  - By Using Synchronized Method
  - By Using Synchronized Block
  - By Using Static Synchronization

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**The synchronized keyword is applicable for whom?** A. Variables and Methods B. Methods and Classes C. Variables and Classes D. Methods and Blocks
Ans.	
Q. 2	**Which Synchronization concept is required when?** A. Multiple threads operating simultaneously on the same object. B. Multiple threads are operating on multiple objects. C. Both (a) and (b) D. None of these
Ans.	
Q. 3	**An object can be locked by how many threads at a time?** A. One. B. Two. C. Zero D. Multiple
Ans.	
Q. 4	**What is the valid syntax for synchronized blocks to get the lock of the current object?** A. synchronized(this). B. synchronized(super). C. synchronized(Test.java) D. None of these
Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=Kni0HMvq1Ig&list=PLbRMhDVUMngcx5xHChJ-f7ofxZI4JzuQR&index=29
2. https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.html
3. https://www.javatpoint.com/synchronization-in-java

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date

Practical No:32	Develop programs to create, write, modify, read operations on Text files.

A. Objective:

Create java programs to create and manage text files.

B. Expected Program Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the Information Technology (IT) problems.
2. **Problem analysis:** Identify and analyze well-defined Information Technology (IT) problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for well-defined Information Technology (IT) technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern Information Technology (IT) tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Develop a program using object-oriented programming concepts with Java to solve the given problems.'

- Problem solving skills by storing and managing programs data into text files.

D. Expected CourseOutcomes (COs):

CO-4: Develop an object-oriented program handling a Text file.

E. PracticalOutcome

After completion of this practical student will be able to,

1. Create a text files to store programs data.
2. Write, modify, and read data of text files.

F. Expected Affective domain Outcome(ADOs)

1. Write a program code by following standard programming rules.
2. Handle computer systems carefully with safety and necessary precaution.
3. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

In Java, with the help of File Class, we can work with files. This File Class is inside the java.io package. The File class can be used by creating an object of the class and then specifying the name of the file.

Why File Handling is Required?

- File Handling is an integral part of any programming language as file handling enables us to store the output of any particular program in a file and allows us to perform certain operations on it.

- In simple words, file handling means reading and writing data to a file.Java Synchronization is better option where we want to allow only one thread to access the shared resource.
- In Java, the concept Stream is used in order to perform I/O operations on a file. So at first, let us get acquainted with a concept known as Stream in Java.

Streams in Java

In Java, a sequence of data is known as a stream.This concept is used to perform I/O operations on a file.

There are two types of streams:

Input Stream:

- The Java InputStream class is the superclass of all input streams. The input stream is used to read data from numerous input devices like the keyboard, network, etc. InputStream is an abstract class, and because of this, it is not useful by itself. However, its subclasses are used to read data.
- There are several subclasses of the InputStream class, which are as follows:
  - AudioInputStream
  - ByteArrayInputStream
  - FileInputStream
  - FilterInputStream
  - StringBufferInputStream
  - ObjectInputStream

Output Stream:

- The output stream is used to write data to numerous output devices like the monitor, file, etc. OutputStream is an abstract superclass that represents an output stream. OutputStream is an abstract class and because of this, it is not useful by itself. However, its subclasses are used to write data.
- There are several subclasses of the OutputStream class which are as follows:
  - ByteArrayOutputStream
  - FileOutputStream
  - StringBufferOutputStream
  - ObjectOutputStream
  - DataOutputStream
  - PrintStream

Based on the data type, there are two types of streams:

Byte Stream:

This stream is used to read or write byte data. The byte stream is again subdivided into two types which are as follows:

- Byte Input Stream: Used to read byte data from different devices.
- Byte Output Stream: Used to write byte data to different devices.

Character Stream:

This stream is used to read or write character data. Character stream is again subdivided into 2 types which are as follows:

- Character Input Stream: Used to read character data from different devices.
- Character Output Stream: Used to write character data to different devices.

H. Experimental set up/ Program Logic-Flow chart:

N.A.

I. Resources Required

S. No.	Instrument/Components / Software	Specification	Quantity
1.	Computer System	Operating System: Any Operating System Processor: At least 1GHz processor Physical Memory: 128 MB or higher Disk space required: 256 MB for JDK	NA
2.	Java Development Kit	JDK 8 or higher.	NA

J. Safety and necessary Precautionsfollowed

1. Do not force components into computer ports.
2. Use genuine software.
3. Use comments to make the program more readable.
4. Turn off the power and unplug equipment after completion of the laboratory session.

K. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

L. Observations and Calculations/Input-Output:

……

……

……

……

……

……

……

……

……

……

……

……

……

M. Practical related Quiz/Questions.

Q. 1	**Which of these class contains the methods used to write in a file?** A. FileStream B. FileInputStream C. BUfferedOutputStream D. FileBufferStream
Ans.	
Q. 2	**Which of these exception is thrown in cases when the file specified for writing is not found?** A. IOException B. FileException C. FileNotFoundException D. FileInputException
Ans.	
Q. 3	**Which of these values is returned by read() method is end of file (EOF) is encountered?** A. 0 B. 1 C. -1 D. Null
Ans.	

N. References / Suggestions

1. https://www.youtube.com/watch?v=lYuj9nn0PBU&list=PLbRMhDVUMngcx5xHChJ-f7ofxZI4JzuQR&index=31
2. https://www.geeksforgeeks.org/file-handling-in-java/
3. https://www.javatpoint.com/file-operations-in-java

O. Assessment-Rubrics

Criteria	Total Marks	Exceptional (5-Marks)	Satisfactory (4 to 3 -Marks)	Developing (2-Marks)	Limited (1-Mark)
Engagement	5	Performed Practical him / her self	Performed Practical with others help	Watched other students performing practical but not tried him / her self	Presenting practical session but not attentively participated in performance
Accuracy	5	Accurately done	1-2Errors/mistakes found	3-5Errors/mistakes identified	More than 5 Errors /mistakes committed
Documentation	5	No errors, Program is well Executed and Documented Properly.	Complete write-up and output tables but presentation is poor	Some of the commands missing with missing outputs	Poor write-up and diagram or missing content
Understanding & Explanation	5	Fully understood the performance & can explain perfectly	Understood the performance but cannot explain	Partially understood the performance & can give little explanation	Partially understood and cannot give explanation
Time	5	Completed the work within 1 week	Work is submitted later than 1st week but by the end of 2nd week	Work done after 2nd week but before the end of 3rd week	Work submitted after 3rd week time

Sign with Date