

# Degree Engineering

A Laboratory Manual for

## Object Oriented Programming (3140705)

### B.E. Semester 4 (Computer)

Enrolment No	
Name	
Branch	
Academic Term	
Institute Name	



**Directorate of Technical Education, Gandhinagar,  
Gujarat**

## **Vishwakarma Government Engineering College Certificate**

*This is to certify that Mr./Ms. \_\_\_\_\_*

*Enrollment No. \_\_\_\_\_ of B.E. Semester 4, Computer Engineering of this Institute  
(GTU Code: 07) has satisfactorily completed the Practical / Tutorial work for the subject  
Object Oriented Programming 1 for the academic year 2022-23.*

Place: \_\_\_\_\_

Date: \_\_\_\_\_

*Name and Sign of Faculty member*

*Head of the Department*

## Preface

Gujarat Technological University (GTU) offers a range of technical courses across various engineering and technology disciplines. These courses are designed to provide students with a comprehensive understanding of theoretical concepts and practical applications in their respective fields. The practicals of courses at GTU involve hands-on experiments, projects, and assignments that allow students to apply theoretical knowledge to real-world situations. Students work in laboratory settings to develop their problem-solving and critical thinking skills while gaining practical experience in their field of study.

GTU has a well-designed curriculum that includes both practical and theory in technical courses. The curriculum is regularly updated to ensure that it stays relevant and provides students with the skills they need to succeed in their careers. The practical and theory of technical courses are structured in a way that allows students to apply theoretical knowledge to practical situations, providing a holistic approach to learning.

In conclusion, GTU offers a comprehensive curriculum that includes practical and theory across various engineering and technology disciplines. The practical provide students with hands-on experience and allow them to apply theoretical knowledge to real-world situations, while the theory provide students with a strong foundation in fundamental concepts and theoretical knowledge in their respective fields. Together, the Computer Engineering Course in GTU provide students with a well-rounded education and prepare them for successful careers in their chosen field.

By using this lab manual students can go through the relevant theory and procedure in advance before the actual performance which creates an interest and students can have basic idea prior to performance. This in turn enhances pre-determined outcomes amongst students. Each experiment in this manual begins with competency, industry relevant skills, course outcomes as well as practical outcomes (objectives). The students will also achieve safety and necessary precautions to be taken while performing practical.

This lab manual is organized in a way that makes it easy for students to follow along with the exercises and examples. Each lab contains a set of objectives, detailed theory/instructions, and exercises to practice the concepts learned. The labs also include review questions to help reinforce key concepts and prepare students for exams.

Java is one of the most popular programming languages in the world, and it is widely used in a variety of industries, including software development, web development, and mobile app development. By learning Core Java in this subject, students will gain a solid foundation in programming concepts and develop the skills they need to build robust, efficient, and scalable applications.

This lab manual covers a wide range of topics, including basic syntax, object-oriented programming concepts, data structures, and exception handling, multi threading etc. It also includes an introduction to GUI programming with javafx.

### **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 technical professionals

### **Institute's Vision**

- To create an ecosystem for proliferation of socially responsible and technically sound engineers, innovators and entrepreneurs.

### **Institute's Mission**

- To develop state-of-the-art laboratories and well-equipped academic infrastructure.
- To motivate faculty and staff for qualification up-gradation, and enhancement of subject knowledge.
- To promote research, innovation and real-life problem-solving skills.
- To strengthen linkages with industries, academic and research organizations.
- To reinforce concern for sustainability, natural resource conservation and social responsibility.

### **Department's Vision**

- To create an environment for providing value-based education in Computer Engineering through innovation, team work and ethical practices.

### **Department's Mission**

- To produce computer engineering graduates according to the needs of industry, government, society and scientific community.
- To develop state of the art computing facilities and academic infrastructure.
- To develop partnership with industries, government agencies and R & D organizations for knowledge sharing and overall development of faculties and students.
- To solve industrial, governance and societal issues by applying computing techniques.
- To create environment for research and entrepreneurship.

## Programme Outcomes (POs)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

### **Program Specific Outcomes (PSOs)**

- Sound knowledge of fundamentals of computer science and engineering including software and hardware.
- Develop the software using sound software engineering principles having web based/mobile based interface.
- Use various tools and technology supporting modern software frameworks for solving problems having large volume of data in the domain of data science and machine learning.

### **Program Educational Objectives (PEOs)**

- Possess technical competence in solving real life problems related to Computing.
- Acquire good analysis, design, development, implementation and testing skills to formulate simple computing solutions to the business and societal needs.
- Provide requisite skills to pursue entrepreneurship, higher studies, research, and development and imbibe high degree of professionalism in the fields of computing.
- Embrace life-long learning and remain continuously employable.
- Work and excel in a highly competence supportive, multicultural and professional environment which abiding to the legal and ethical responsibilities.

**Practical – Course Outcome matrix**

<b>Course Outcomes (COs):</b>						
<b>CO_3140705.1</b>	Use various Java constructs, features and libraries for simple problems.					
<b>CO_3140705.2</b>	Demonstrate how to define and use classes, interfaces, create objects and methods, how to override and overload methods, compile and execute programs.					
<b>CO_3140705.3</b>	Write a program using exception handling, multithreading with synchronization.					
<b>CO_3140705.4</b>	Write a program using Files, binary I/O, collection Frameworks for a given problem.					
<b>CO_3140705.5</b>	Design and develop GUI based applications in a group using modern tools and frameworks.					
<b>Sr. No.</b>	<b>Practical Outcome/Title of experiment</b>	<b>CO 1</b>	<b>CO 2</b>	<b>CO 3</b>	<b>CO 4</b>	<b>CO 5</b>
1.	1.1. Write a Program that displays Welcome to Java, Learning Java Now and Programming is fun. 1.2. Write a program that solves the following equation and displays the value x and y: 1) $3.4x + 50.2y = 44.5$ 2) $2.1x + .55y = 5.9$ (Assume Cramer's rule to solve equation $ax + by = e$ $x = \frac{ed - bf}{ad - bc}$ $cx + dy = f$ $y = \frac{af - ec}{ad - bc}$ ) 1.3. Write a program that reads a number in meters, converts it to feet, and displays the result. 1.4. Write a program that prompts the user to enter three integers and display the integers in decreasing order.	√				
2.	2.1. Write a program for sorting an elements in array, and enter elements by command line argument. 2.2. write a program to implement grow able stack. 2.3. Write a Program to print: 0 1 2 3 4 5 6 7 8 9 Using two dimension array. 2.4. Write a program for calculator to accept an expression as a string in which the operands and operator are separated by spaces. For ex: $3 + 4$ .	√				
3.	3.1. Write a program to perform method overloading of area method to find area of rectangle and square 3.2. Do above program using constructor overloading. 3.3. Create a class student with necessary properties, methods and constructor. Overload a function name search in this class which allows us to search student based on roll number, name and city.		√			

4.	<p>4.1 write a program that create following : a inherited by b ,b inherited by c and c is inherited by d. a has one variable b has two and c has 3 variables. Use constructor and super to initialized the variables of classes.</p> <p>4.2 Write a Java program that combine several classes and interfaces the abstract class robot has subclass name robotA, robotB, robotC. Class robotA1 extends robotA. Class robotB1 and robotB2 extends robotB. Class robotC1 extends robotC. The locomotion interface declares three methods has forward , reverse and stop. It is implemented by class robotB and robotC. The sound interface declares one method named beep and it is implemented by robotA1,robotB1 and robotC. Defined all class and implement interface as specified than invoked beep method of all class object that are of type sound.</p> <p>4.3 write a program that has abstract class dim which has a two variable dim1,dim2. triangle and rectangle are subclass of dim. implement dynamic method dispatch by creating reference of dim.</p>		√			
5.	<p>5.1 Write a java program to find solution of quadratic equation. Take care of divide by zero error and other arithmetic exceptions</p> <p>5.2 Write a program in which main() call a(),a() call b(),b() call c(),c() call d(),d() generates ArrayIndexOutOfBoundsException. c and d catch throw the Exception. b() handle an Exception. Each method contain finally.</p> <p>5.3 Write a program to get value of radius through keyboard and calculate area of circle. Take care of InputMismatchException.</p> <p>5.4 Write a program to create an array of 10 integers. Get value of those 10 integers using console. Now ask for an index of array through keyboard then divide the array into two from that index. Take care of array index out of bound exception. Also handle InputMismatchException.</p> <p>5.5 Create a class name student which stores information like roll number, name, phone number, address, course etc. Write a function which accepts an object of student to add a new student in existing list of student. While adding check for roll number. The roll number should be in 3 digit. Implement this check using user define exception class</p> <p>5.6 Write the bin2Dec (string binary String) method to convert a binary string into a decimal number. Implement the bin2Dec method to throw a NumberFormatException if the string is not a binary string.</p>		√			



6.	<p>6.1 Write a Program for multithreading that finds prime number in the range provided by user in command line argument depending upon range creates sufficient number of child thread.</p> <p>6.2 Write a program that demonstrate thread priority four threads each with a different priority level then the other are started objects and not the behave of each Thread</p> <p>6.3 Write a program that demonstrate use of Executor Framework in multitasking.</p> <p>6.4 Write a program for handling producer consumer problem</p>			√		
7.	<p>7.1 Write a program that creates an Array List and adds a Loan object , a Date object , a string, and a Circle object to the list, and use a loop to display all elements in the list by invoking the object's toString() method.</p> <p>7.2 Using list perform following operation on it in java program. (use ArrayList and LinkedList )</p> <p>7.2.1 Creating a new list</p> <p>7.2.2 Basic operations.</p> <p>7.2.3 Iterating over a list</p> <p>7.2.4 Searching for an element in a list</p> <p>7.2.5 Sorting a list</p> <p>7.2.6 Copying one list into another</p> <p>7.2.7 Shuffling elements in a list</p> <p>7.2.8 Reversing elements in a list</p> <p>7.2.9 Extracting a portion of a list</p> <p>7.2.10 Converting between Lists and arrays</p> <p>7.2.11 List to Stream</p> <p>7.2.12 Concurrent lists.</p> <p>7.3 Write a java program to evaluate arithmetic operation using stack.</p> <p>7.4 Implement a java program to show various operation of queue.</p> <p>7.5 Define MYPriorityQueue class that extends Priority Queue to implement the Cloneable interface and implement the clone() method to clone a priority queue</p> <p>7.6 Create a generic class GenericStack&lt;E&gt; with all methods of stack. And create two object, first object is for String values and second object is for Double values.</p> <p>7.7 Create a Generic Method sort which will sort an array of <b>Comparable</b> objects. The objects are instances of the <b>Comparable</b> interface, and they are compared using the <b>compareTo</b> method. To test the method, the program sorts an array of integers, an array of double numbers, an array of characters, and an array of strings.</p>			√		

	7.8 In prg 7.7 create a newclass which has method max() that will take GenericStack<> object as argument using bounded wildcard (? Extends T) where T can be of type Number. And it will find max number from stack.					
8.	<p>8.1 Demonstrate test time and removed time required for hash set,tree set,array set,linked list.</p> <p>8.2 Write a Java program to find the longest word in a text file.</p> <p>8.3 Write a program to create a file name 123.txt, if it does not exist. Append a new data to it if it already exist. write 150 integers created randomly into the file using Text I/O. Integers are separated by space.</p> <p>8.4 Write a program that reads words from a text file and displays all the nonduplicate words in descending order.The text file is passed as a command-line argument .</p> <p>8.5 Write a program to find whether a String is palindrome or not by using recursive helper method.</p>				√	
9.	<p>Implemet Following using Java FX</p> <p>9.1 Write a program that moves a circle up, down, left or right using arrow keys.</p> <p>9.2 Write a program that displays the color of a circle as red when the mouse button is pressed and as blue when the mouse button is released.</p> <p>9.3 Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed.</p>					√
10.	Develop an simple java application which uses Java FX.					√

## Industry Relevant Skills

The following industry relevant competency are expected to be developed in the student by undertaking the practical work of this laboratory.

1. Will be able to start primary JAVA programming.
2. Will be able to use various IDEs like Netbeans/Eclipse.
3. Will be able to create GUI based small application using javafx
4. Will be able to understand Advance java in future semester.

## Guidelines for Faculty members

1. Teacher should provide the guideline with demonstration of practical to the students with all features.
2. Teacher shall explain basic concepts/theory related to the experiment to the students before starting of each practical
3. Involve all the students in performance of each experiment.
4. Teacher is expected to share the skills and competencies to be developed in the students and ensure that the respective skills and competencies are developed in the students after the completion of the experimentation.
5. Teachers should give opportunity to students for hands-on experience after the demonstration.
6. Teacher may provide additional knowledge and skills to the students even though not covered in the manual but are expected from the students by concerned industry.
7. Teacher is expected to refer complete curriculum of the course and follow the guidelines for implementation.

## Instructions for Students

1. Students are expected to carefully listen to all the theory classes delivered by the faculty members and understand the COs, content of the course, teaching and examination scheme, skill set to be developed etc.
2. Students shall understand basic concept and then implement it in program.
3. Students shall develop logical skill as expected by IT industries.
4. Student shall attempt to develop Programming skills and build confidence.
5. Student shall develop the habits of evolving more ideas, innovations, skills etc. apart from those included in scope of manual.
6. Student should develop a habit of submitting the experimentation work as per the schedule and s/he should be well prepared for the same.
7. Student should develop a habit of working in team.

## Common Safety Instructions

Students are expected to

- 1) Switch on the PC carefully (not to use wet hands)
- 2) Shutdown the PC properly at the end of your Lab
- 3) Carefully Handle the peripherals (Mouse, Keyboard, Network cable etc)
- 4) Use Laptop in lab after getting permission from Teacher

## Index (Progressive Assessment Sheet)

Sr. No.	Objective(s) of Experiment	PG No.	Date of performance	Date of submission	Assessment Marks	Sign. of Teacher with date	Remarks
1	<p>1.5. Write a Program that displays Welcome to Java, Learning Java Now and Programming is fun.</p> <p>1.6. Write a program that solves the following equation and displays the value x and y:            1) <math>3.4x + 50.2y = 44.5</math> 2) <math>2.1x + .55y = 5.9</math>            (Assume Cramer's rule to solve equation <math>ax + by = e</math> <math>x = \frac{ed - bf}{ad - bc}</math> <math>cx + dy = f</math> <math>y = \frac{af - ec}{ad - bc}</math>)</p> <p>1.7. Write a program that reads a number in meters, converts it to feet, and displays the result.</p> <p>1.8. Write a program that prompts the user to enter three integers and display the integers in decreasing order.</p>						
2	<p>2.1. Write a program for sorting an elements in array, and enter elements by command line argument.</p> <p>2.2. write a program to implement grow able stack.</p> <p>2.3. Write a Program to print:            0            1 2            3 4 5            6 7 8 9            Using two dimension array.</p> <p>2.4. Write a program for calculator to accept an expression as a string in which the operands and operator are separated by spaces. For ex: <math>3 + 4</math>.</p>						
3	<p>3.1. Write a program to perform method overloading of area method to find area of rectangle and square</p> <p>3.2. Do above program using constructor overloading.</p> <p>3.3. Create a class student with necessary properties, methods and constructor. Overload a function name search in this class which allows us to search student based on roll number, name and city.</p>						
4	4.1 write a program that create following : a inherited by b ,b inherited by c and c is						

	<p>inherited by d. a has one variable b has two and c has 3 variables. Use constructor and super to initialize the variables of classes.</p> <p>4.2 Write a Java program that combines several classes and interfaces. The abstract class robot has subclass name robotA, robotB, robotC. Class robotA1 extends robotA. Class robotB1 and robotB2 extends robotB. Class robotC1 extends robotC. The locomotion interface declares three methods: forward, reverse, and stop. It is implemented by class robotB and robotC. The sound interface declares one method named beep and it is implemented by robotA1, robotB1, and robotC. Define all class and implement interface as specified then invoke beep method of all class object that are of type sound.</p> <p>4.3 Write a program that has abstract class dim which has two variables dim1, dim2. Triangle and rectangle are subclass of dim. Implement dynamic method dispatch by creating reference of dim.</p>					
5	<p>5.1 Write a Java program to find solution of quadratic equation. Take care of divide by zero error and other arithmetic exceptions.</p> <p>5.2 Write a program in which main() calls a(), a() calls b(), b() calls c(), c() calls d(), d() generates ArrayIndexOutOfBoundsException. c and d catch and throw the Exception. b() handles an Exception. Each method contains finally.</p> <p>5.3 Write a program to get value of radius through keyboard and calculate area of circle. Take care of InputMismatchException.</p> <p>5.4 Write a program to create an array of 10 integers. Get value of those 10 integers using console. Now ask for an index of array through keyboard then divide the array into two from that index. Take care of array index out of bound exception. Also handle InputMismatchException.</p> <p>5.5 Create a class name student which stores information like roll number, name, phone number, address, course etc. Write a function which accepts an object of student to add a new student in existing list of student. While adding check for roll number. The roll number should be in 3 digit. Implement this check using user-defined exception class.</p>					

	5.6 Write the bin2Dec (string binary String) method to convert a binary string into a decimal number. Implement the bin2Dec method to throw a NumberFormatException if the string is not a binary string.					
6	<p>6.1 Write a Program for multithreading that finds prime number in the range provided by user in command line argument depending upon range creates sufficient number of child thread.</p> <p>6.2 Write a program that demonstrate thread priority four threads each with a different priority level then the other are started objects and not the behave of each Thread</p> <p>6.3 Write a program that demonstrate use of Executor Framework in multitasking.</p> <p>6.4 Write a program for handling producer consumer problem</p>					
7	<p>7.1 Write a program that creates an Array List and adds a Loan object , a Date object , a string, and a Circle object to the list, and use a loop to display all elements in the list by invoking the object's toString() method.</p> <p>7.2 Using list perform following operation on it in java program. (use ArrayList and LinkedList )</p> <p>7.2.1 Creating a new list</p> <p>7.2.2 Basic operations.</p> <p>7.2.3 Iterating over a list</p> <p>7.2.4 Searching for an element in a list</p> <p>7.2.5 Sorting a list</p> <p>7.2.6 Copying one list into another</p> <p>7.2.7 Shuffling elements in a list</p> <p>7.2.8 Reversing elements in a list</p> <p>7.2.9 Extracting a portion of a list</p> <p>7.2.10 Converting between Lists and arrays</p> <p>7.2.11 List to Stream</p> <p>7.2.12 Concurrent lists.</p> <p>7.3 Write a java program to evaluate arithmetic operation using stack.</p> <p>7.4 Implement a java program to show various operation of queue.</p> <p>7.5 Define MYPriorityQueue class that extends Priority Queue to implement the Cloneable interface and implement the clone() method to clone a priority queue</p> <p>7.6 Create a generic class GenericStack&lt;E&gt; with all methods of stack. And create two object, first object is for String values and second object is for Double values.</p>					

	<p>7.7 Create a Generic Method sort which will sort an array of <b>Comparable</b> objects. The objects are instances of the <b>Comparable</b> interface, and they are compared using the <b>compareTo</b> method. To test the method, the program sorts an array of integers, an array of double numbers, an array of characters, and an array of strings.</p> <p>7.8 In prg 7.7 create a newclass which has method max() that will take GenericStack&lt;&gt; object as argument using bounded wildcard (? Extends T) where T can be of type Number. And it will find max number from stack.</p>						
8	<p>8.1 Demonstrate test time and removed time required for hash set,tree set,an array set,linked list.</p> <p>8.2 Write a Java program to find the longest word in a text file.</p> <p>8.3 Write a program to create a file name 123.txt, if it does not exist. Append a new data to it if it already exist. write 150 integers created randomly into the file using Text I/O. Integers are separated by space.</p> <p>8.4 Write a program that reads words from a text file and displays all the nonduplicate words in descending order.The text file is passed as a command-line argument .</p> <p>8.5 Write a program to find whether a String is palindrome or not by using recursive helper method.</p>						
9	<p>Implemet Following using Java FX</p> <p>9.1 Write a program that moves a circle up, down, left or right using arrow keys.</p> <p>9.2 Write a program that displays the color of a circle as red when the mouse button is pressed and as blue when the mouse button is released.</p> <p>9.3 Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed.</p>						
10	Develop an simple java application which uses Java FX.						
Total							

# Practical No: 1

**Aim:**

- 1.1. Write a Program that displays Welcome to Java, Learning Java Now and Programming is fun.
- 1.2. Write a program that solves the following equation and displays the value x and y:  
1)  $3.4x + 50.2y = 44.5$  2)  $2.1x + .55y = 5.9$   
(Assume Cramer's rule to solve equation  
 $ax + by = e$   $x = \frac{ed - bf}{ad - bc}$   
 $cx + dy = f$   $y = \frac{af - ec}{ad - bc}$  )
- 1.3. Write a program that reads a number in meters, converts it to feet, and displays the result.
- 1.4. Write a program that prompts the user to enter three integers and display the integers in decreasing order.

**1.1. Write a Program that displays Welcome to Java, Learning Java Now and Programming is fun.****Date:****Competency and Practical Skills: Programming****Relevant CO: CO1**

**Objectives:** (a) To understand java's byte code and method of running java program by setting environment variables 'path' and 'classpath' in command prompt.

**Equipment/Instruments: jdk 11, notepad****Theory:**

Java's bytecode is a set of instructions that is generated by the Java compiler when it compiles a Java program. These instructions are machine-readable and platform-independent, meaning they can be executed on any platform that has a Java Virtual Machine (JVM) installed.

When you write a Java program, you write it in the Java programming language, which is a high-level language. The Java compiler takes your high-level code and translates it into bytecode, which is a low-level, machine-readable code. This bytecode can be executed on any platform that has a JVM installed, making Java a platform-independent language.

The JVM is responsible for interpreting the bytecode and executing the instructions. It takes the bytecode, interprets it, and executes the instructions on the underlying hardware. This means that the same Java program can be run on any platform that has a JVM installed, without any changes to the code.

Java bytecode is often used in server-side applications, web applications, and mobile applications. It is also used in the development of desktop applications, games, and other software.

java and Javac are two separate executables in Java development.



Java is the executable that runs Java programs. When you execute a Java program, the Java virtual machine (JVM) reads the bytecode generated by the Java compiler and executes the instructions on the underlying hardware.

Javac, on the other hand, is the executable that compiles Java programs. It takes the source code written in Java and translates it into bytecode that can be executed by the JVM.

To compile a Java program using Javac, you need to open a command prompt or terminal, navigate to the directory where the Java file is located, and run the Javac command followed by the name of the Java file. For example:

```
javac myfirstprg.java
```

This will compile the MyProgram.java file and generate a bytecode file named myfirstprg.class. You can then run the program using the Java command:

```
java myfirstprg
```

Steps to run java program on command prompt:

2. Install java or just copy jdk folder in drive(assume c:\).
3. Create a folder(assume java2023) in any drive(assume d:\) for storing all programs.
4. Open command prompt.
5. Go to your created folder and drive.
6. Set path variable to the path of bin folder: "set path=c:\jdk11\bin\"
7. Set class path to the folder you made : "set classpath=d:\java2023\"
8. Compile using javac exe: javac filename.jav
9. Run class file using :java filename"

Program:

```
class myfirstprg
{
    Public static void main(String arg[])
    {
        System.out.println("Welcome to java");
        System.out.println("Learning Java Now");
        System.out.println("Programming is fun");
    }
}
```

### Safety and necessary Precautions:

Java must be in System and set proper environment variables.  
Also remember that you are compiling java file and running class file .

### Observations:

Output:

```
Welcome to java
Learning Java Now
Programming is fun
```

**1.2 Write a program that solves the following equation and displays the value x and y:**

1)  $3.4x + 50.2y = 44.5$  2)  $2.1x + .55y = 5.9$   
(Assume Cramer's rule to solve equation  
 $ax + by = e$   $x = \frac{ed - bf}{ad - bc}$   
 $cx + dy = f$   $y = \frac{af - ec}{ad - bc}$  )

**Date: ://Write date of experiment here**

**Competency and Practical Skills: Programming**

**Relevant CO: CO1**

**Objectives:**

(a) To understand java's Data type and operators.

**Equipment/Instruments: jdk11 or above**

**Theory:**

Data types: Java has two categories of data types: primitive data types and reference data types.

Primitive data types are built-in data types that are part of the Java language. There are eight primitive data types in Java:

1. byte - 8-bit signed integer
2. short - 16-bit signed integer
3. int - 32-bit signed integer
4. long - 64-bit signed integer
5. float - 32-bit floating-point number
6. double - 64-bit floating-point number
7. boolean - true or false
8. char - 16-bit Unicode character

Reference data types are objects that are created from classes. They are more complex than primitive data types, and they require more memory to store. Examples of reference data types include:

1. String - a sequence of characters
2. Arrays - a collection of values of the same data type
3. Classes - a blueprint for creating objects

In addition to the primitive and reference data types, Java also has a special data type called "void." This data type is used to indicate that a method does not return a value.

It's important to note that Java is a strongly-typed language, which means that each variable and expression must have a declared data type. This helps prevent common errors that can occur when working with data.

Java has several operators that can be used to perform different operations on variables and values. Some of the most commonly used operators in Java include:

1. **Arithmetic Operators:** These are used to perform arithmetic operations on variables and values. The arithmetic operators in Java include + (addition), - (subtraction), \* (multiplication), / (division), and % (modulus).
2. **Assignment Operators:** These are used to assign values to variables. The assignment operators in Java include = (simple assignment), += (add and assign), -= (subtract and assign), \*= (multiply and assign), /= (divide and assign), and %= (modulus and assign).
3. **Comparison Operators:** These are used to compare values and variables. The comparison operators in Java include == (equal to), != (not equal to), > (greater than), < (less than), >= (greater than or equal to), and <= (less than or equal to).
4. **Logical Operators:** These are used to combine multiple conditions and return a boolean value. The logical operators in Java include && (logical AND), || (logical OR), and ! (logical NOT).
5. **Bitwise Operators:** These are used to perform bitwise operations on binary values. The bitwise operators in Java include & (bitwise AND), | (bitwise OR), ^ (bitwise XOR), ~ (bitwise NOT), << (left shift), >> (right shift), and >>> (unsigned right shift).
6. **Ternary Operator:** This operator is used to assign a value to a variable based on a condition. The ternary operator in Java is represented by the ? : symbol.

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

- 1.3 Write a program that reads a number in meters, converts it to feet, and displays the result.  
1.4 Write a program that prompts the user to enter a letter and check whether a letter is vowel or constant.

**Date: //Write date of experiment here**

**Competency and Practical Skills: Programming**

**Relevant CO: CO1**

**Objectives:**

- (a) To understand input in java.

**Equipment/Instruments: jdk11 or above**

**Theory:**

In Java, the Scanner class is used to read input from the user through the console. Here are the basic steps to read input using Scanner in Java:

1. Import the Scanner class at the beginning of your program using the following line of code:

```
import java.util.Scanner;
```

2. Create an instance of the Scanner class by using the following line of code:

```
Scanner scanner = new Scanner(System.in);
```

3. Use the various methods of the Scanner class to read input from the user. The most commonly used methods include:

- `nextLine()`: Reads a line of text entered by the user until they press the Enter key. Returns the input as a String.
- `nextInt()`: Reads an integer entered by the user. Returns the input as an int.
- `nextDouble()`: Reads a floating-point number entered by the user. Returns the input as a double.

**Procedure:**

**//Write programs here(1.3 & 1.4)**

**Observations:**

**//Write program output here**

- 1.5 Write a program that prompts the user to enter three integers and display the integers in decreasing order.

**Date: //Write date of experiment here**

**Competency and Practical Skills: Programming**

**Relevant CO: CO1**

**Objectives**

- (a) To understand java's Control Statements.

**Equipment/Instruments: jdk 11,notepad.****Theory:**

Control statements are used to control flow of code. Basic control statements in java are almost similar to c/c++.

Selection statements:

if ..else,switch etc.

Iteration statements:

For,while,do...while etc

Jump statements:

Break,break with label,continue and continue with label.

**Safety and necessary Precautions:**

- a. Solve compile time errors

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion:** This experiment is used to understand basic structure of core java. And student can start programming in java.

**Quiz:**

1. Difference between OOP and POP.
2. Write characteristics of OOP.
3. Write characteristics of java.
4. Describe java's Bytecode.

**Suggested Reference:**

1. Intro to Java Programming, 10<sup>th</sup> edition, Y.Daniel Liang, Pearson
2. Complete reference Core JAVA by Herbert Schildt. MC Graw Hill.

**References used by the students:**

1. Topics of Introduction to java, java's bytecode, operators, control statement from Complete reference Core JAVA by Herbert Schildt. MC Graw Hill.

**Rubric wise marks obtained:**

Rubrics	Knowledge (2)		Problem Recognition(2)		Logic Building (2)		Completeness and accuracy (2)		Ethics (2)		Total
	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	
Marks											

## Practical -2

**Aims:**

- 2.5. Write a program for sorting an elements in array, and enter elements by command line argument.
- 2.6. write a program to implement grow able stack.
- 2.7. Write a Program to print:  
0  
1 2  
3 4 5  
6 7 8 9  
Using two dimension array.
- 2.8. Write a program for calculator to accept an expression as a string in which the operands and operator are separated by spaces. For ex: 3 + 4.

**Competency and Practical Skills: Programming****Relevant CO: CO1**

**Objectives:** (a) To understand java's array, array operations ,basics of String .

**Equipment/Instruments: jdk 11 , Notepad.**

**Theory:****Array:**

Java support single dimension and multi dimension arrays.

Single dimension array:

Syntax:

```
type vari_name[];  
var_name=new type[size];
```

Or combine

```
type var_name[]=new type[];
```

Multi dimension array:

In java multi dimension array is an array of arrays. When you allocate memory for a multidimensional array, you need only specify the memory for the first (leftmost) dimension. You can allocate the remaining dimensions separately.

For Ex:

```
int twoD[][] = new int[4][];  
twoD[0] = new int[5];  
twoD[1] = new int[5];  
twoD[2] = new int[5];  
twoD[3] = new int[5];
```

**String:**

The first thing to understand about strings is that every string you create is actually an object of type String.

The second thing to understand about strings is that objects of type String are immutable; once a String object is created, its contents cannot be altered.

For Ex:

```
String myString = "this is a test";  
System.out.println(myString);
```

Basic methods :

```
boolean equals(String object)  
int length()  
char charAt(int index)
```

### Programs:

**2.1. Write a program for sorting an elements in array, and enter elements by command line argument.**

**Date: ://Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**2.2 write a program to implement grow able stack.**

**Date: ://Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**2.3 Write a Program to print:**

```
0  
1 2  
3 4 5  
6 7 8 9  
Using two dimension array.
```

**Date: ://Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**2.4. Write a program for calculator to accept an expression as a string in which the Operands and operator are separated by spaces. For ex: 3 + 4.**

**Date: ://Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion:** This experiment is used to understand array and String.

**Quiz:**

2. Describe two dimension Array in java.
3. Describe Short Circuit Operators in java( && and ||)
4. Describe >> and << operators.
5. Describe any five methods of String class.

**Suggested Reference:**

1. Intro to Java Programming, 10<sup>th</sup> edition, Y.Daniel Liang, Pearson
2. Complete reference Core JAVA by Herbert Schildt. MC Graw Hill.

**References used by the students:**

**//Write reference used by you**

**Rubric wise marks obtained:**

Rubrics	Knowledge (2)		Problem Recognition( 2)		Logic Building (2)		Completeness and accuracy (2)		Ethics (2)		Total
	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	
Marks											



## Practical- 3

### Aims:

- 3.1. Write a program to perform method overloading of area method to find area of rectangle and square
- 3.2. Do above program using constructor overloading
- 3.3. Create a class student with necessary properties, methods and constructor. Overload a function name search in this class which allows us to search student based on roll number, name and city.

### Relevant CO: CO2

### Objectives:

To understand and implement method class, object , Method overloading , constructor overloading

**Equipment/Instruments: jdk 11 , notepad.**

### Theory:

#### Class and Object :

#### Syntax:

```
class classname {  
    type instance-variable1;  
    type instance-variable2;  
    // ...  
    type instance-variableN;  
    type methodname1(parameter-list) {  
        // body of method  
    }  
    type methodname2(parameter-list) {  
        // body of method  
    }  
    // ...  
    type methodnameN(parameter-list) {  
        // body of method  
    }  
}
```

### Object:

The **new** operator dynamically allocates memory for an object. It has this general form:

```
Class name class-var;  
class-var = new classname( );
```

### Constructor:

A constructor initializes an object immediately upon creation. It has the same name as the class in which it resides and is syntactically similar to a method. Once defined, the constructor is automatically called immediately after the object is created, before the **new** operator completes. Constructors look a little strange because they have no return type, not even **void**. This is because the implicit return type of a class' constructor is the class type itself.

Syntax:

```
classname()  
{  
//Body of Constructor;  
}
```

For ex:

```
Box mybox1 = new Box();
```

Here mybox1 is class box object. **new Box( )** is calling the **Box( )** constructor. When you do not explicitly define a constructor for a class, then Java creates a default constructor for the class.

We can also pass parameter to constructor:

For ex:

```
class Box  
{  
double depth;  
Box(double d)  
{  
depth=d;  
}  
}
```

We can create object as

```
Box ob=new Box(10.5);
```

### Overloading:

Method overloading is an example of Polymorphism. Methods can have same name but it differ in terms of arguments either number of arguments are different or types of arguments are different.

For Ex:

```
void test(int a)  
{  
}  
void test(int a,int b)  
{  
}
```

Constructor overloading can also be implemented in same way as method overloading.

For ex:

```
Box(double d)  
{  
}  
Box(double w , double d)  
{  
}
```

**Programs:**

**3.1. Write a program to perform method overloading of area method to find area of rectangle and square**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**3.2. Do above program using constructor overloading**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**3.3. Create a class student with necessary properties, methods and constructor. Overload a function name search in this class which allows us to search student based on roll number, name and city.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion:** This experiment is used to understand and implement class, object, constructor, Method and constructor Overloading.

**Quiz:**

1. Describe Method Overloading in java.
2. Describe class and object in java

**Suggested Reference:**

1. Intro to Java Programming, 10<sup>th</sup> edition, Y. Daniel Liang, Pearson
2. Complete reference Core JAVA by Herbert Schildt. MC Graw Hill.

**References used by the students:**

**//Write your reference here**

**Rubric wise marks obtained:**

Rubrics	Knowledge (2)		Problem Recognition(2)		Logic Building (2)		Completeness and accuracy (2)		Ethics (2)		Total
	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	
Marks											

## Practical- 4

### Aims:

- 4.1 write a program that create following : a inherited by b ,b inherited by c and c is inherited by d. a has one variable b has two and c has 3 variables. Use constructor and super to initialized the variables of classes.
- 4.2 Write a Java program that combine several classes and interfaces the abstract class robot has subclass name robotA, robotB, robotC. Class robotA1 extends robotA. Class robotB1 and robotB2 extends robotB. Class robotC1 extends robotC. The locomotion interface declares three methods has forward , reverse and stop. It is implemented by class robotB and robotC. The sound interface declares one method named beep and it is implemented by robotA1,robotB1 and robotC. Defined all class and implement interface as specified than invoked beep method of all class object that are of type sound.
- 4.3 write a program that has abstract class dim which has a two variable dim1,dim2. triangle and rectangle are subclass of dim. implement dynamic method dispatch by creating reference of dim.

### Relevant CO: CO2

### Objectives:

To understand and implement Inheritance, abstract class ,Method Overriding and Dynamic method dispatched and interface.

**Equipment/Instruments:** jdk 11 , notepad ,

### Theory:

#### Inheritance:

Object-oriented programming allows you to define new classes from existing classes. This is called inheritance. it allows the creation of hierarchical classifications. Using inheritance, you can create a general class that defines traits common to a set of related items. This class can then be inherited by other, more specific classes, each adding those things that are unique to it. In the terminology of Java, a class that is inherited is called a *superclass*. The class that does the inheriting is called a *subclass*. Therefore, a subclass is a specialized version of a superclass. 'extends' keyword is used for inheriting super class.

Use of 'super':

1. Used to call super class constructor.
2. Used to access variables of super class(like this)

#### Syntax:

**Class subclassname extends superclassname**

```
{  
}
```

#### Method Overrriding:

A subclass inherits methods from a superclass. Sometimes it is necessary for the subclass to modify the implementation of a method defined in the superclass. This is referred to as method overriding. So subclass has method with same name and signature as in superclass but the code is different. It is different than Method overloading.

Overloading means to define multiple methods with the same name but different signatures. Overriding means to provide a new implementation for a method in the subclass.

**Dynamic binding/Dynamic method dispatched:**

Superclass reference variable can refer to subclass object. This fact is used to implement runtime polymorphism. We can create superclass object and assign subclass object to it. When we call overridden method using super class object, it will call the version of method written in that subclass which object is assigned to superclass object.

**Abstract class and Method:**

Sometimes Methods written in superclass may not have meanings. But we have to write it in super class in order to override it. Such method can be declared as abstract method abstract method has no body in super class. The class that contains at least a abstract method must be declared as an abstract class. We can not create object of abstract class but we can create reference of it.

For Ex:

```
abstract class Dimension
{
    abstract double area();
}
```

**Interface:**

In Java, an interface is a type that defines a set of method signatures that a class must implement if it wants to use that interface. An interface is similar to a class, but it cannot be instantiated on its own, and all of its methods must be implemented by a class that implements the interface.

To define an interface in Java, the "interface" keyword is used, followed by the name of the interface and the method signatures that it defines.

For example:

```
public interface MyInterface {
    void myMethod();
    int myOtherMethod(String input);
}
```

Classes that implement an interface must use the "implements" keyword and provide an implementation for all of the methods defined in the interface.

For example:

```
public class MyClass implements MyInterface
{
    public void myMethod() {
        // Implementation goes here
    }

    public int myOtherMethod(String input) {
        // Implementation goes here
    }
}
```

Interfaces can also be used to achieve polymorphism, allowing multiple classes that implement the same interface to be used interchangeably.

**Programs:**

- 4.1. write a program that create following : a inherited by b ,b inherited by c and c is inherited by d. a has one variable b has two and c has 3 variables. Use constructor and super to initialized the variables of classes.

Date: **//Write date of experiment here**

Procedure:

**//Write program here**

Observations:

**//Write program output here**

- 4.2. Write a Java program that combine several classes and interfaces the abstract class robot has subclass name robotA, robotB, robotC. Class robotA1 extends robotA. Class robotB1 and robotB2 extends robotB. Class robotC1 extends robotC. The locomotion interface declares three methods has forward , reverse and stop. It is implemented by class robotB and robotC. The sound interface declares one method named beep and it is implemented by robotA1,robotB1 and robotC. Defined all class and implement interface as specified than invoked beep method of all class object that are of type sound.

Date: **//Write date of experiment here**

Procedure:

**//Write program here**

Observations:

**//Write program output here**

- 4.3. write a program that has abstract class dim which has a two variable dim1,dim2. triangle and rectangle are subclass of dim. implement dynamic method dispatch by creating reference of dim.

Date: **//Write date of experiment here**

Procedure:

**//Write program here**

Observations:

**//Write program output here**

Conclusion: **//Write conclusion here.**

**Quiz:**

1. Compare Method Overloading and Method Overriding.
2. Describe Dynamic Method Dispatched.
3. Write about abstract Class and Method.

**Suggested Reference:**

3. Intro to Java Programming, 10<sup>th</sup> edition, Y.Daniel Liang, Pearson
4. Complete reference Core JAVA by Herbert Schildt. MC Graw Hill.

**References used by the students:**

**//Write your reference here**

**Rubric wise marks obtained:**

Rubrics	Knowledge (2)		Problem Recognition(2)		Logic Building (2)		Completeness and accuracy (2)		Ethics (2)		Total
	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	
<b>Marks</b>											



## Practical- 5

### Aims:

- 5.7 Write a java program to find solution of quadratic equation. Take care of divide by zero error and other arithmetic exceptions
- 5.8 Write a program in which main() call a(),a() call b(),b() call c(),c() call d(),d() generates ArrayIndexOutOfBoundsException. c and d catch throw the Exception. b() handle an Exception. Each method contain finally.
- 5.9 Write a program to get value of radius through keyboard and calculate area of circle. Take care of InputMismatchException.
- 5.10 Write a program to create an array of 10 integers. Get value of those 10 integers using console. Now ask for an index of array through keyboard then divide the array into two from that index. Take care of array index out of bound exception. Also handle InputMismatchException.
- 5.11 Create a class name student which stores information like roll number, name, phone number, address, course etc. Write a function which accepts an object of student to add a new student in existing list of student. While adding check for roll number. The roll number should be in 3 digit. Implement this check using user define exception class
- 5.12 Write the bin2Dec (string binary String) method to convert a binary string into a decimal number. Implement the bin2Dec method to throw a NumberFormatException if the string is not a binary string.

### Relevant CO: CO3

**Objectives:** To understand and implement Exception handling in java.

**Equipment/Instruments:** jdk 11 , Any IDE.

### Theory:

An exception is a run-time error. A Java exception is an object that describes an exceptional (that is, error) condition that has occurred in a piece of code. When an exceptional condition arises, an object representing that exception is created and thrown in the method that caused the error. That method may choose to handle the exception itself, or pass it on. Either way, at some point, the exception is caught and processed.

Java exception handling is managed via five keywords: try, catch, throw, throws, and finally

Program statements that you want to monitor for exceptions are contained within a try block. If an exception occurs within the try block, it is thrown. Your code can catch this exception (using catch) and handle it in some rational manner. System-generated exceptions are automatically thrown by the Java run-time system. To manually throw an exception, use the keyword throw. Any exception that is thrown out of a method must be specified as such by a throws clause. Any code that absolutely must be executed before a method returns is put in a finally block.

### Syntax:

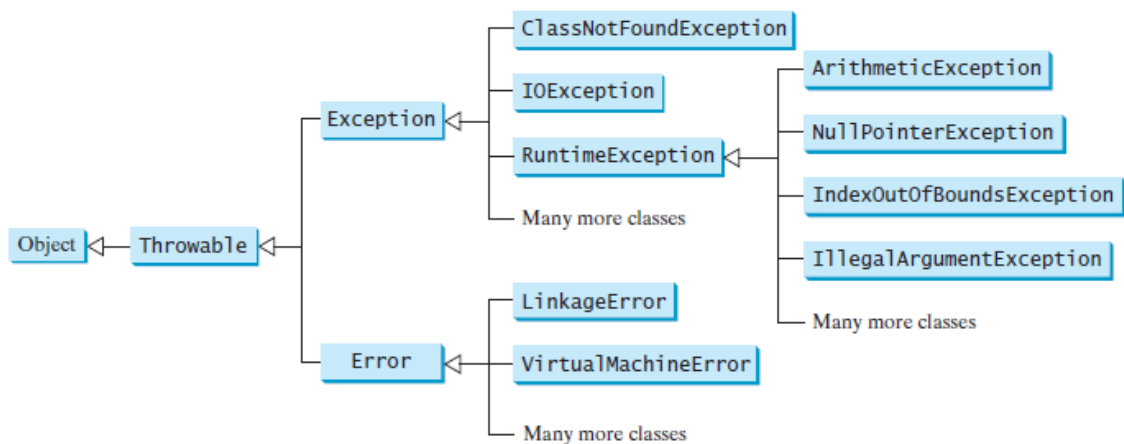
```
try {  
    // block of code to monitor for errors  
}
```

```

catch (ExceptionType1 exOb) {
// exception handler for ExceptionType1
}
catch (ExceptionType2 exOb) {
// exception handler for ExceptionType2
}
finally {
// block of code to be executed before try block ends
}

```

Class Hierarchy[from reference book]:



**Programs:**

**5.1 Write a java program to find solution of quadratic equation. Take care of divide by zero error and other arithmetic exceptions**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**5.2. Write a program in which main() call a(),a() call b(),b() call c(),c() call d(),d() generates ArrayIndexOutOfBoundsException. c and d catch throw the Exception. b() handle an Exception. Each method contain finally.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**5.3 Write a program to get value of radius through keyboard and calculate area of circle. Take care of InputMismatchException.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**5.4 Write a program to create an array of 10 integers. Get value of those 10 integers using console. Now ask for an index of array through keyboard then divide the array into two from that index. Take care of array index out of bound exception. Also handle InputMismatchException.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion: //Write conclusion here.**

**5.5 Create a class name student which stores information like roll number, name, phone number, address, course etc. Write a function which accepts an object of student to add a new student in existing list of student. While adding check for roll number. The roll number should be in 3 digit. Implement this check using user define exception class.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion: //Write conclusion here.**

**5.6 Write the bin2Dec (string binary String) method to convert a binary string into a decimal number. Implement the bin2Dec method to throw a NumberFormatException if the string is not a binary string.**

**Date: //Write date of experiment here**

**Procedure:****//Write program here****Observations:****//Write program output here****Conclusion: //Write conclusion here.****Quiz:**

1. Define Exception. Explain Exception handling in java.
2. What is checked and unchecked Exception.
3. Describe chained Exception.

**Suggested Reference:**

1. Intro to Java Programming, 10<sup>th</sup> edition, Y.Daniel Liang, Pearson
2. Complete reference Core JAVA by Herbert Schildt. MC Graw Hill.

**References used by the students:****//Write your reference here****Rubric wise marks obtained:**

Rubrics	Knowledge (2)		Problem Recognition( 2)		Logic Building (2)		Completeness and accuracy (2)		Ethics (2)		Total
	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	
Marks											

## Practical- 6

### Aims:

- 6.1 Write a Program for multithreading that finds prime number in the range provided by user in command line argument depending upon range creates sufficient number of child thread.
- 6.2 Write a program that demonstrate thread priority four threads each with a different priority level then the other are started objects and not the behave of each Thread
- 6.3 Write a program that demonstrate use of Executor Framework in mutitasking.
- 6.4 Write a program for handling producer consumer problem.

### Relevant CO: CO3

### Objectives:

To understand and implement Multithreading in java.

**Equipment/Instruments: jdk 11 , notepad.**

### Theory:

A program may consist of many tasks that can run concurrently. A thread is the flow of execution, from beginning to end, of a task.

A thread provides the mechanism for running a task. With Java, you can launch multiple threads from a program concurrently. These threads can be executed simultaneously in multiprocessor Systems

Thread can be created in two ways:

1. By implementing Runnable interface.
2. By extending Thread class.

### Thread pools:

Starting a new thread for each task could limit throughput and cause poor performance. Using a thread pool is an ideal way to manage the number of tasks executing concurrently. Java provides the **Executor** interface for executing tasks in a thread pool and the **ExecutorService** interface for managing and controlling tasks. **ExecutorService** is a subinterface of **Executor**

### Programs:

**6.1. Write a Program for multithreading that finds prime number in the range provided by user in command line argument depending upon range creates sufficient number of child thread.**

**Date: //Write date of experiment here**

### Procedure:

**//Write program here**

### Observations:

**//Write program output here**

**6.2. Write a program that demonstrate thread priority four threads each with a different priority level then the other are started objects and not the behave of each Thread**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**6.3. Write a program that demonstrate use of Executor Framework in mutitasking.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion: //Write conclusion here.**

**6.4 Write a program for handling producer consumer problem.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion: //Write conclusion here.**

**Quiz:**

**Suggested Reference:**

1. Intro to Java Programming, 10<sup>th</sup> edition, Y.Daniel Liang, Pearson
2. Complete reference Core JAVA by Herbert Schildt. MC Graw Hill.

**References used by the students:**

**//Write your reference here**

**Rubric wise marks obtained:**

Rubrics	Knowledge (2)		Problem Recognition( 2)		Logic Building (2)		Completeness and accuracy (2)		Ethics (2)		Total
	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	
Marks											

## Practical- 7

### Aims:

- 7.1 Write a program that creates an Array List and adds a Loan object , a Date object , a string, and a Circle object to the list, and use a loop to display all elements in the list by invoking the object's to String() method.
- 7.2 Using list perform following operation on it in java program. (use ArrayList and LinkedList )
  - 7.2.1 Creating a new list
  - 7.2.2 Basic operations.
  - 7.2.3 Iterating over a list
  - 7.2.4 Searching for an element in a list
  - 7.2.5 Sorting a list
  - 7.2.6 Copying one list into another
  - 7.2.7 Shuffling elements in a list
  - 7.2.8 Reversing elements in a list
  - 7.2.9 Extracting a portion of a list
  - 7.2.10 Converting between Lists and arrays
  - 7.2.11 List to Stream
  - 7.2.12 Concurrent lists.
- 7.3 Write a java program to evaluate arithmetic operation using stack.
- 7.4 Implement a java program to show various operation of queue.
- 7.5 Define MYPriorityQueue class that extends Priority Queue to implement the Cloneable interface and implement the clone() method to clone a priority queue
- 7.6 Create a generic class GenericStack<E> with all methods of stack. And create two object, first object is for String values and second object is for Double values.
- 7.7 Create a Generic Method sort which will sort an array of **Comparable** objects. The objects are instances of the **Comparable** interface, and they are compared using the **compareTo** method. To test the method, the program sorts an array of integers, an array of double numbers, an array of characters, and an array of strings.
- 7.8 In prg 7.7 create a newclass which has method max() that will take GenericStack<> object as argument using bounded wildcard (? Extends T) where T can be of type Number. And it will find max number from stack.

**Relevant CO: CO4**

### Objectives:

**To make use of ArrayList class and other classes.**

**To understand and implements List, Stacks, Queues ,Priority Queues etc**

**To understand and use Generics.**

**Equipment/Instruments: jdk 11 , Any IDE**



**Theory:****ArrayList:**

An ArrayList object can be used to store a list of objects.

Now we are ready to introduce a very useful class for storing objects. You can create an array to store objects. But, once the array is created, its size is fixed. Java provides the ArrayList class, which can be used to store an unlimited number of objects

java.util.ArrayList<E>	
+ArrayList()	Creates an empty list.
+add(o: E): void	Appends a new element o at the end of this list.
+add(index: int, o: E): void	Adds a new element o at the specified index in this list.
+clear(): void	Removes all the elements from this list.
+contains(o: Object): boolean	Returns true if this list contains the element o.
+get(index: int): E	Returns the element from this list at the specified index.
+indexOf(o: Object): int	Returns the index of the first matching element in this list.
+isEmpty(): boolean	Returns true if this list contains no elements.
+lastIndexOf(o: Object): int	Returns the index of the last matching element in this list.
+remove(o: Object): boolean	Removes the first element o from this list. Returns true if an element is removed.
+size(): int	Returns the number of elements in this list.
+remove(index: int): boolean	Removes the element at the specified index. Returns true if an element is removed.
+set(index: int, o: E): E	Sets the element at the specified index.

ArrayList class[1]

Ex:

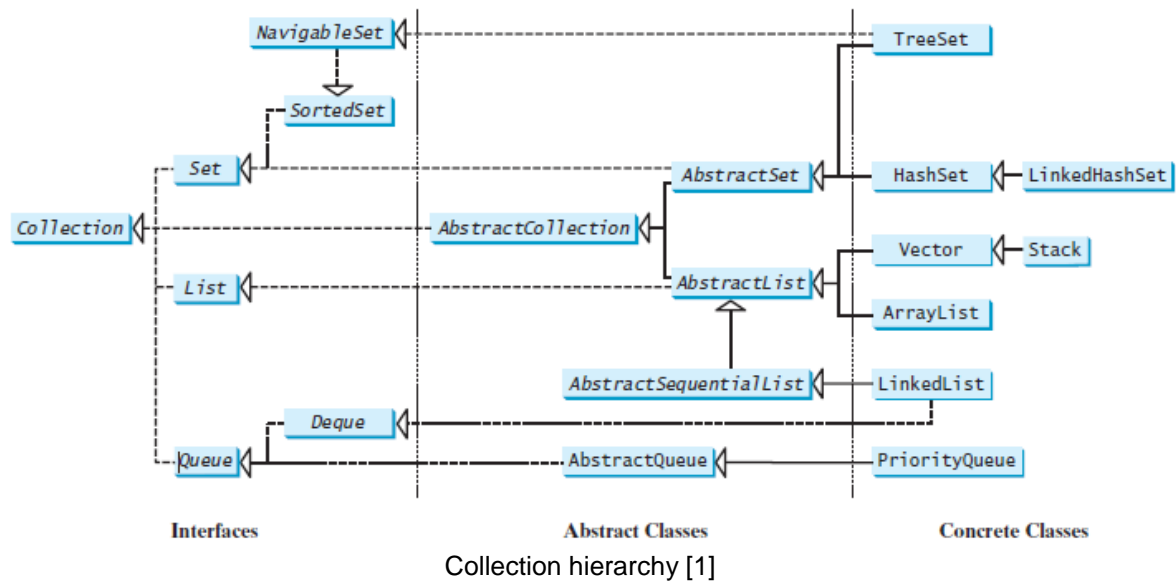
```
ArrayList<String> cities = new ArrayList<String>();
```

**Collection :**

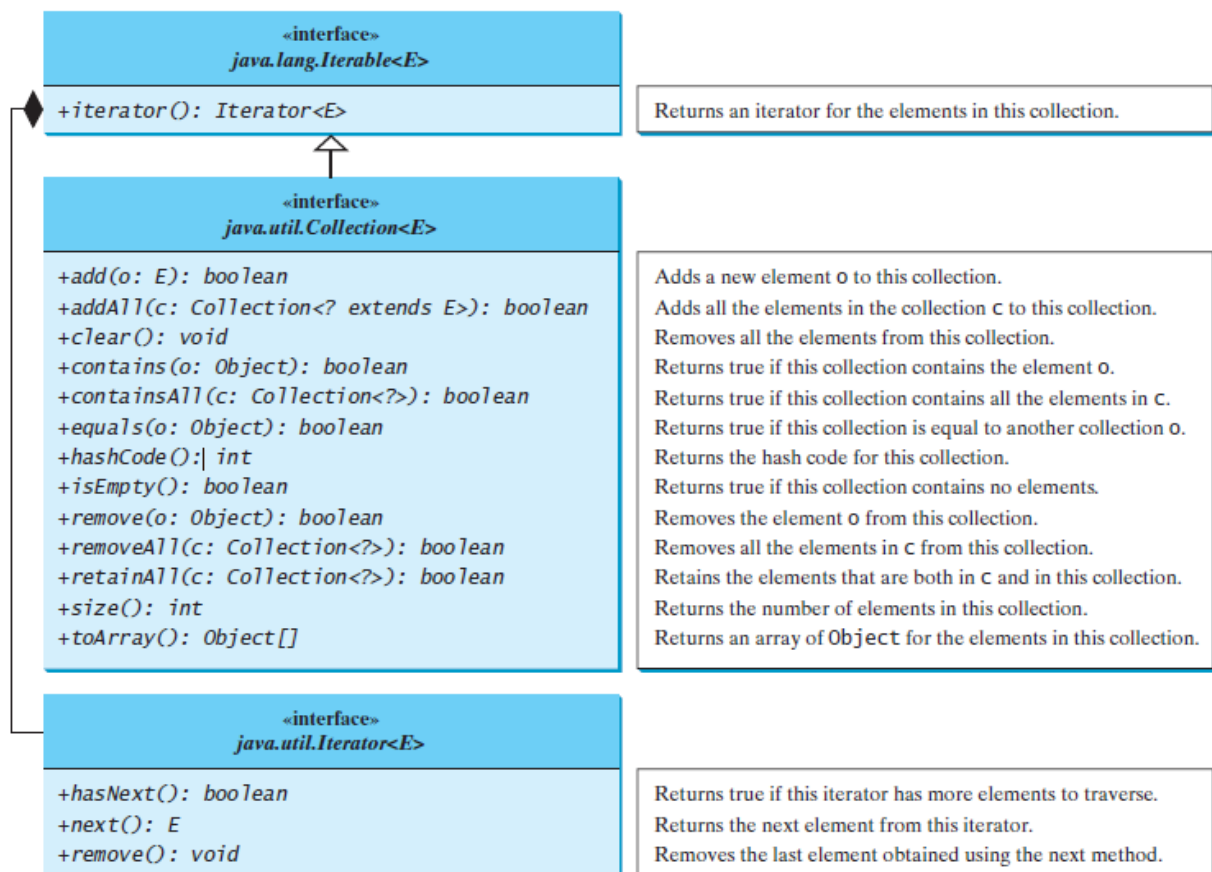
The Collection interface defines the common operations for lists, vectors, stacks, queues, priority queues, and sets.

The Java Collections Framework supports two types of containers:

- One for storing a collection of elements is simply called a collection.
- The other, for storing key/value pairs, is called a map.



### Methods in Collection and Iterator interface:



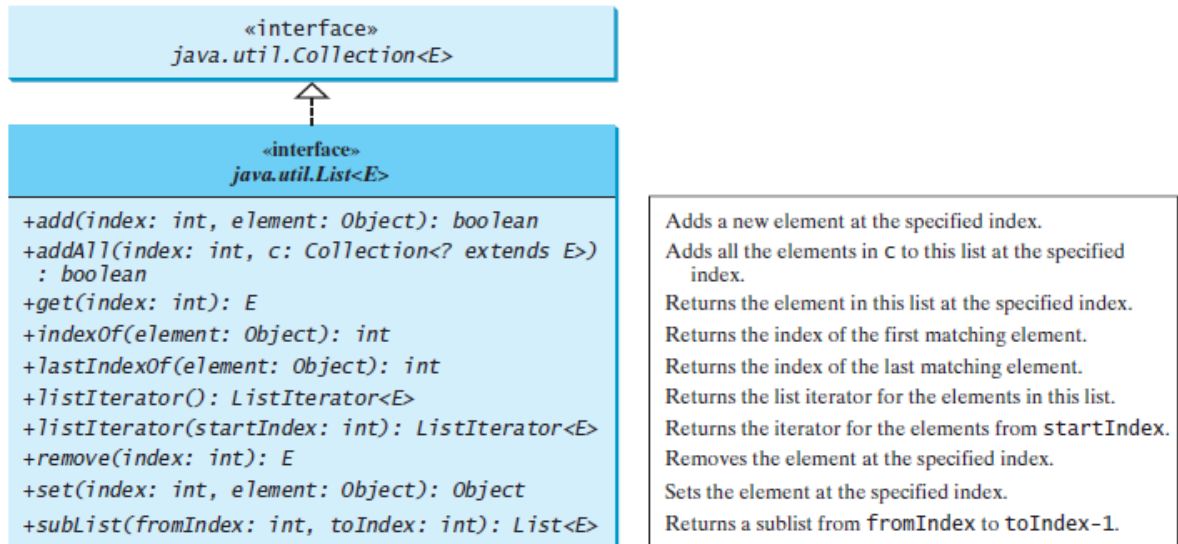
List of Methods[1]

The Collection interface contains the methods for manipulating the elements in a collection, and you can obtain an iterator object for traversing elements in the collection. Each collection is Iterable. You can obtain its Iterator object to traverse all the elements in the collection. Iterator is a classic design pattern for walking through a data structure without having to expose the details of how data is stored in the data structure.

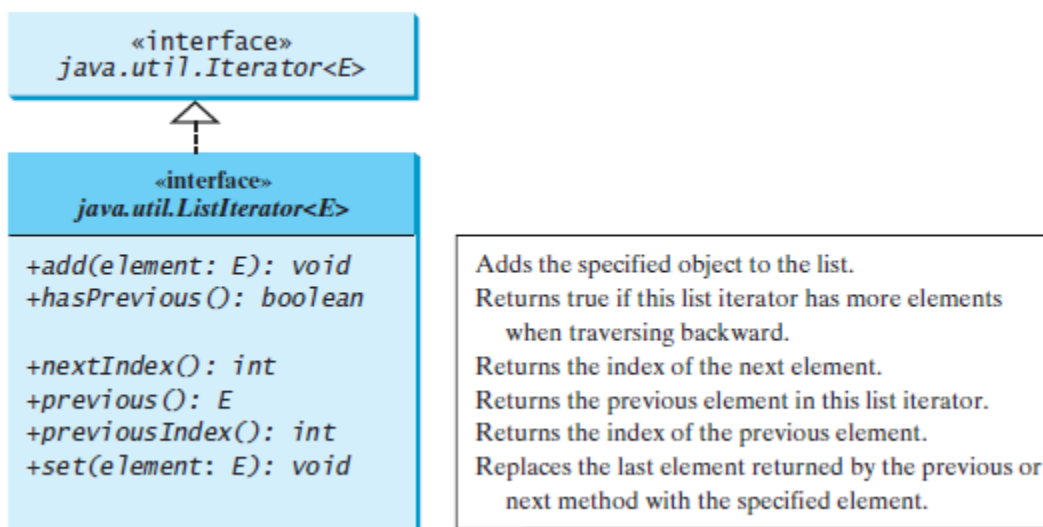
The Collection interface extends the Iterable interface. The Iterable interface defines the iterator method, which returns an iterator. The Iterator interface provides a uniform way for traversing elements in various types of collections.

## Lists:

The List interface extends the Collection interface and defines a collection for storing elements in a sequential order. To create a list, use one of its two concrete classes: ArrayList or LinkedList.



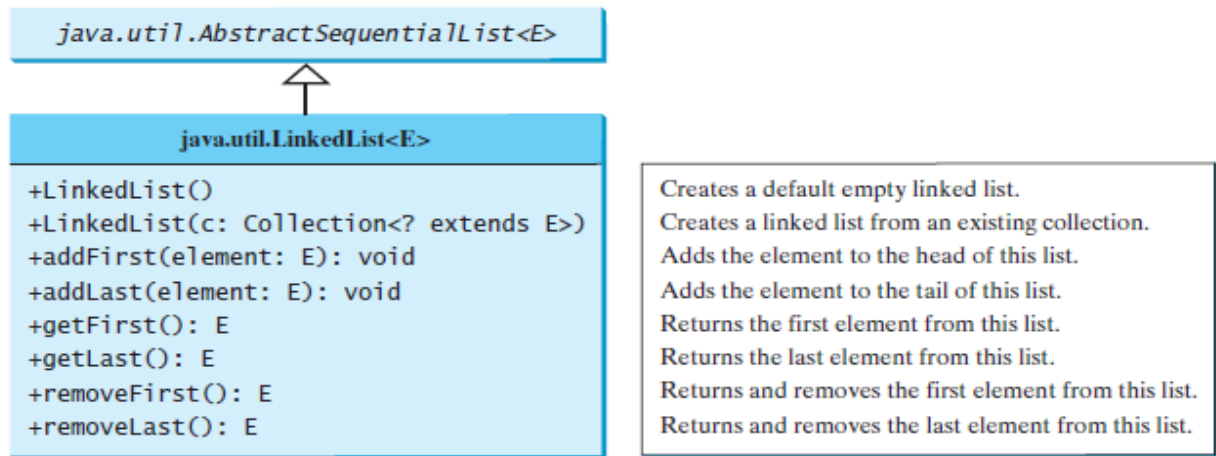
List interface[1]



ListIterator interface[1]

## LinkedList class:

LinkedList is a linked list implementation of the List interface. In addition to implementing the List interface, this class provides the methods for retrieving, inserting, and removing elements from both ends of the list, as shown below[1]:



List class method[1]

**The Comparator interface:**

Comparator can be used to compare the objects of a class that doesn't implement Comparable. Several classes in the Java API, such as String, Date, Calendar, BigInteger, BigDecimal, and all the numeric wrapper classes for the primitive types, implement the Comparable interface. The Comparable interface defines the compareTo method, which is used to compare two elements of the same class that implement the Comparable interface.

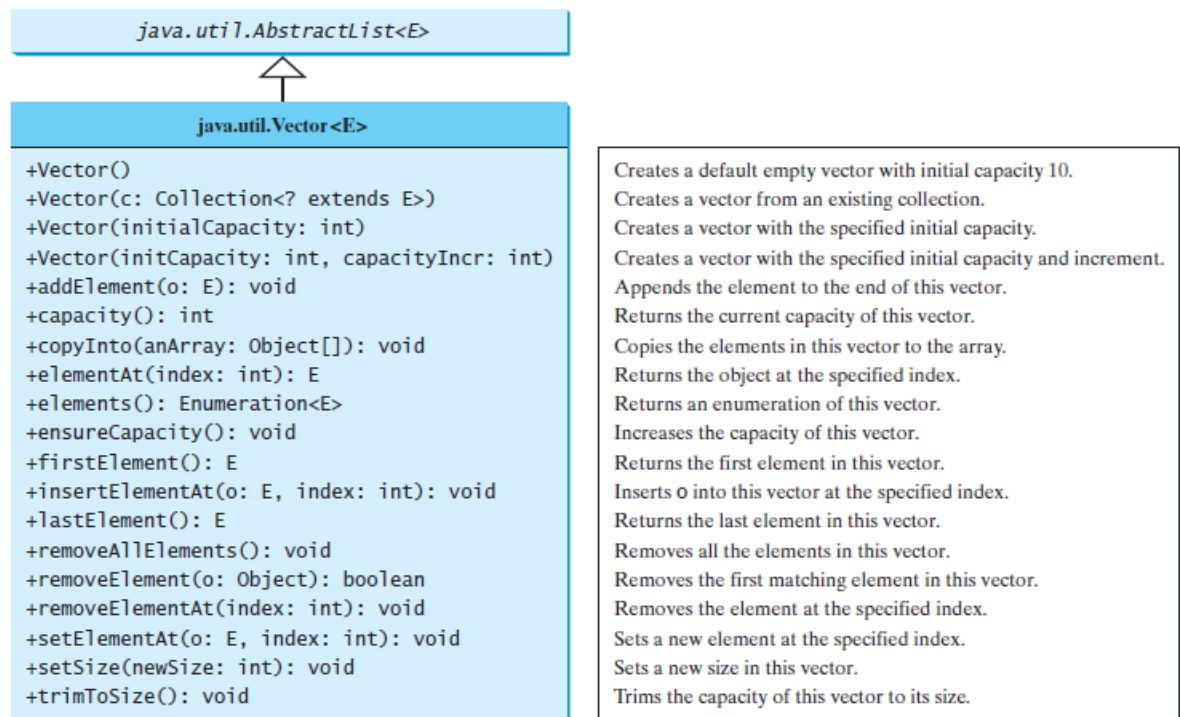
```
public int compare(T element1, T element2)
```

Returns a negative value if element1 is less than element2, a positive value if element1 is greater than element2, and zero if they are equal.

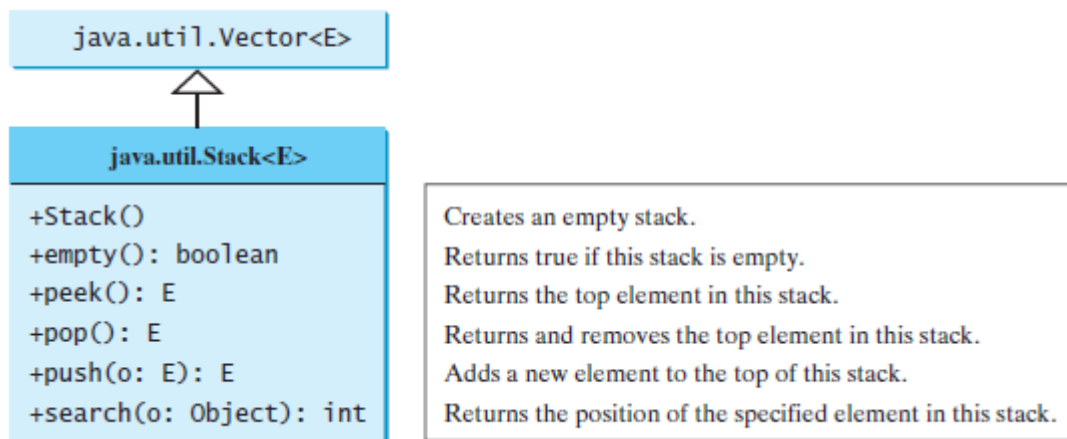
**Vector and Stack classes:**

Vector is a subclass of AbstractList, and Stack is a subclass of Vector in the Java API.

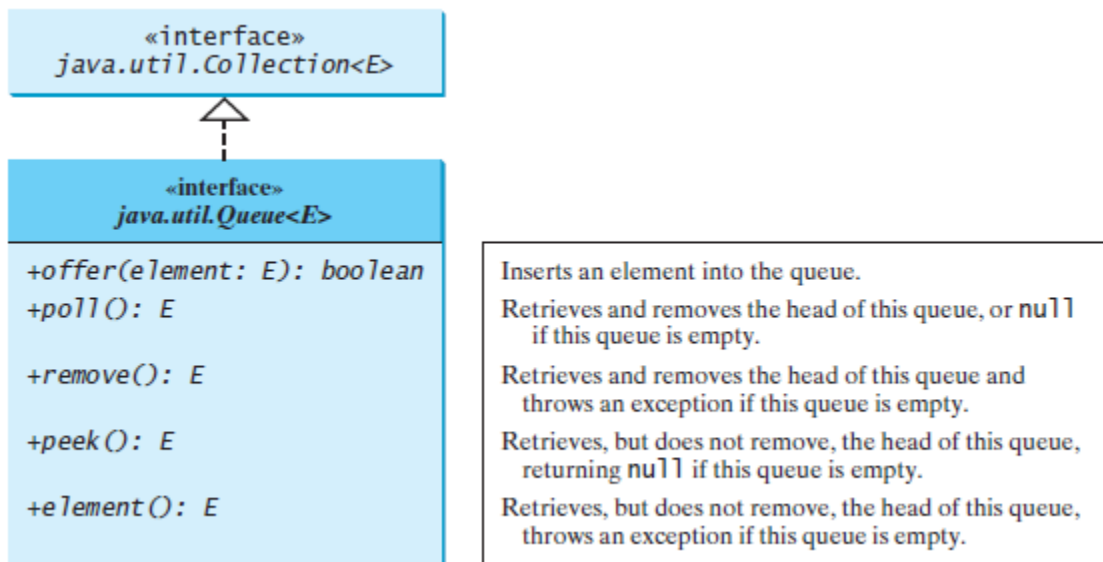
Vector is the same as ArrayList, except that it contains synchronized methods for accessing and modifying the vector. Synchronized methods can prevent data corruption when a vector is accessed and modified by two or more threads concurrently.



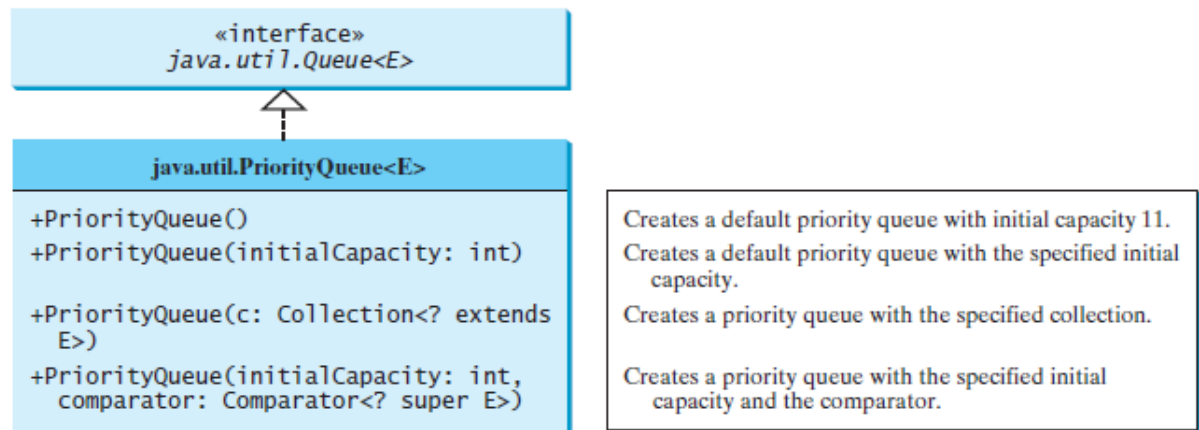
Vector class[1]

**Stack class:**

Stack class[1]

**Queue interface:**

## Queue interface[1]

**Priority Queue:**

Priority Queue class[1]

**Generics:**

Generics enable you to detect errors at compile time rather than at runtime.

Java has allowed you to define generic classes, interfaces, and methods since JDK 1.5. Several interfaces and classes in the Java API were modified using generics. For example, prior to JDK 1.5 the `java.lang.Comparable` interface was defined as shown in Figure 19.1a, but since JDK 1.5 it is modified as shown below:

```
package java.lang;

public interface Comparable {
    public int compareTo(Object o)
}
```

(a) Prior to JDK 1.5

```
package java.lang;

public interface Comparable<T> {
    public int compareTo(T o)
}
```

(b) JDK 1.5

Here, `<T>` represents a formal generic type, which can be replaced later with an actual concrete type. Replacing a generic type is called a generic instantiation. By convention, a single capital letter such as `E` or `T` is used to denote a formal generic type.

**Generic class:**

A generic type can be defined for a class or interface. A concrete type must be specified when using the class to create an object or using the class or interface to declare a reference variable.

For ex:

```
public class GenericStack<E>
{
    private java.util.ArrayList<E> list = new java.util.ArrayList<>();

    public int getSize() {
        return list.size();
    }
}
```

```
public E peek() {  
    return list.get(getSize() - 1);  
}  
  
public void push(E o) {  
    list.add(o);  
}  
  
public E pop() {  
    E o = list.get(getSize() - 1);  
    list.remove(getSize() - 1);  
    return o;  
}  
public boolean isEmpty() {  
    return list.isEmpty();  
}  
  
@Override  
public String toString() {  
    return "stack: " + list.toString();  
}  
}
```

### Generic Method:

A generic type can be defined for a static method. You can define generic interfaces (e.g., the Comparable interface in Figure 19.1b) and classes (e.g., the GenericStack class in Listing 19.1). You can also use generic types to define generic methods. For example, Listing 19.2 defines a generic method print (lines 10–14) to print an array of objects.

To declare a generic method, you place the generic type <E> immediately after the keyword static in the method header.

For example, public static <E> void print(E[] list)

### Wildcard Generic Types:

You can use unbounded wildcards, bounded wildcards, or lower-bound wildcards to specify a range for a generic type.

The first form, ?, called an unbounded wildcard, is the same as ? extends Object. The second form, ? extends T, called a bounded wildcard, represents T or a subtype of T. The third form, ? super T, called a lower-bound wildcard, denotes T or a supertype of T.

### Programs:

**7.1 Write a program that creates an Array List and adds a Loan object , a Date object , a string, and a Circle object to the list, and use a loop to display all elements in the list by invoking the object's toString() method.**

**Date: //Write date of experiment here**

**Procedure:**



**//Write program here**

**Observations:**

**//Write program output here**

**7.2 Using list perform following operation on it in java program. (use ArrayList and LinkedList )**

- 7.2.1 Creating a new list**
- 7.2.2 Basic operations.**
- 7.2.3 Iterating over a list**
- 7.2.4 Searching for an element in a list**
- 7.2.5 Sorting a list**
- 7.2.6 Copying one list into another**
- 7.2.7 Shuffling elements in a list**
- 7.2.8 Reversing elements in a list**
- 7.2.9 Extracting a portion of a list**
- 7.2.10 Converting between Lists and arrays**
- 7.2.11 List to Stream**
- 7.2.12 Concurrent lists.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion: //Write conclusion here.**

**7.3 Write a java program to evaluate arithmetic operation using stack.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion: //Write conclusion here.**

**7.4 Implement a java program to show various operation of queue.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**



**//Write program output here**

**Conclusion: //Write conclusion here.**

**7.5 Define MYPriorityQueue class that extends Priority Queue to implement the Cloneable interface and implement the clone() method to clone a priority queue**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion: //Write conclusion here.**

**7.6 Create a generic class GenericStack<E> with all methods of stack. And create two object, first object is for String values and second object is for Double values.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion: //Write conclusion here.**

**7.7 Create a Generic Method sort which will sort an array of Comparable objects. The objects are instances of the Comparable interface, and they are compared using the compareTo method. To test the method, the program sorts an array of integers, an array of double numbers, an array of characters, and an array of strings.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion: //Write conclusion here.**

**7.8 In practical 7.7 create a newclass which has method max() that will take GenericStack<> object as argument using bounded wildcard (? Extends T) where T can be of type Number. And it will find max number from stack.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here****Observations:****//Write program output here****Conclusion: //Write conclusion here****Quiz:**

1. Write methods of collection.
2. Describe wildcard Generic types with example.
3. Describe ArrayList class with example.

**Suggested Reference:**

1. 1.Intro to Java Programming, 10<sup>th</sup> edition, Y.Daniel Liang, Pearson
2. Complete reference Core JAVA by Herbert Schildt. MC Graw Hill.

**References used by the students:****//Write your reference here****Rubric wise marks obtained:**

Rubrics	Knowledge (2)		Problem Recognition(2)		Logic Building (2)		Completeness and accuracy (2)		Ethics (2)		Total
	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	
Marks											

## Practical- 8

### Aims:

- 8.1 Demonstrate test time and removed time required for hash set, tree set, an array set, linked list.
- 8.2 Write a Java program to find the longest word in a text file.
- 8.3 Write a program to create a file name 123.txt, if it does not exist. Append a new data to it if it already exists. Write 150 integers created randomly into the file using Text I/O. Integers are separated by space.
- 8.4 Write a program that reads words from a text file and displays all the non-duplicate words in descending order. The text file is passed as a command-line argument.
- 8.5 Write a program to find whether a String is a palindrome or not by using a recursive helper method.

### Relevant CO: CO4

### Objectives:

To understand and implement Sets, Maps, and file handling.

### Equipment/Instruments: jdk 11 , Any IDE

### Theory:

#### Sets:

You can create a set using one of its three concrete classes: HashSet, LinkedHashSet, or TreeSet.

The Set interface extends the Collection interface. It does not introduce new methods or constants, but it stipulates that an instance of Set contains no duplicate elements. The concrete classes that implement Set must ensure that no duplicate elements can be added to the set. That is, no two elements  $e_1$  and  $e_2$  can be in the set such that  $e_1.equals(e_2)$  is true.

#### Hash Set and LinkedHashSet:

The HashSet class is a concrete class that implements Set. You can create an empty hash set using its no-arg constructor or create a hash set from an existing collection.

By default, the initial capacity is 16 and the load factor is 0.75. If you know the size of your set, you can specify the initial capacity and load factor in the constructor. Otherwise, use the default setting. The load factor is a value between 0.0 and 1.0.

The load factor measures how full the set is allowed to be before its capacity is increased. When the number of elements exceeds the product of the capacity and load factor, the capacity is automatically doubled. For example, if the capacity is 16 and load factor is 0.75, the capacity will be doubled to 32 when the size reaches 12 ( $16 * 0.75 = 12$ ). A higher load factor decreases the space costs but increases the search time. Generally, the default load factor 0.75 is a good tradeoff between time and space costs.

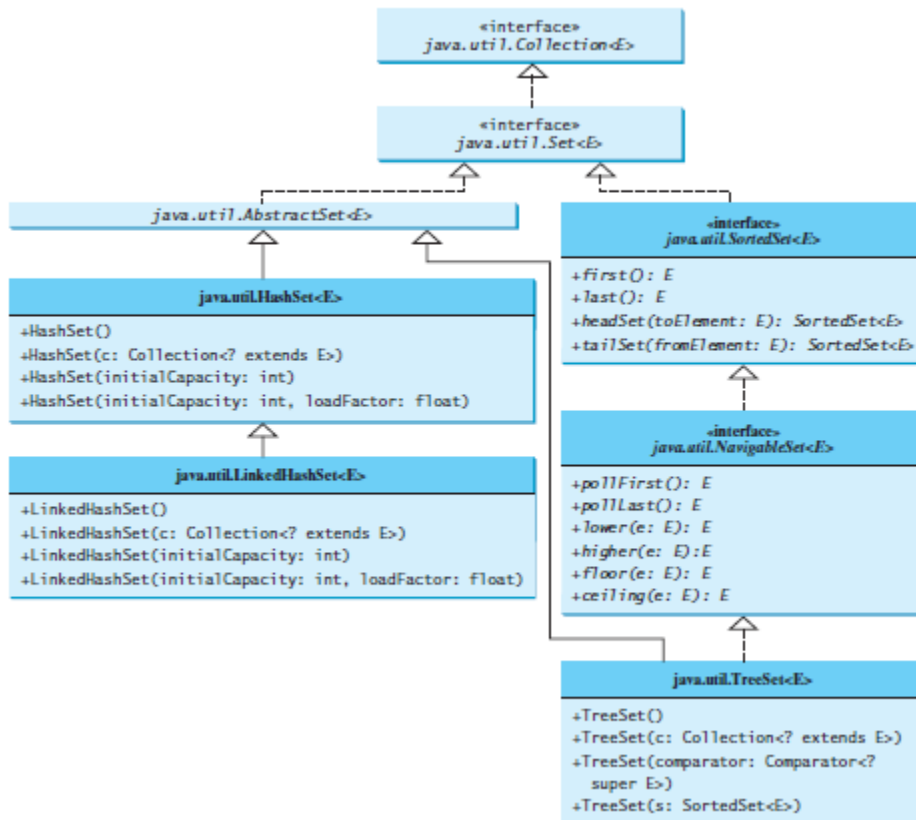
A HashSet can be used to store duplicate-free elements. For efficiency, objects added to a hash set need to implement the hashCode method in a manner that properly disperses the hash code.

Linked Hash Set:

LinkedHashSet extends HashSet with a linked-list implementation that supports an ordering of the elements in the set. The elements in a HashSet are not ordered, but the elements in a LinkedHashSet can be retrieved in the order in which they were inserted into the set. A LinkedHashSet can be created by using one of its four constructors.

### TreeSet:

SortedSet is a subinterface of Set, which guarantees that the elements in the set are sorted. Additionally, it provides the methods first() and last() for returning the first and last elements in the set, and headSet(toElement) and tailSet(fromElement) for returning a portion of the set whose elements are less than toElement and greater than or equal to fromElement, respectively.

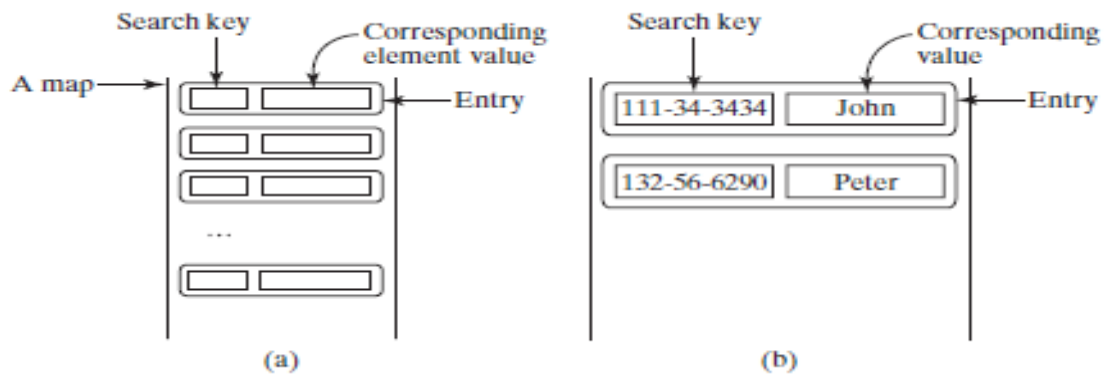


Methods of sets[1]

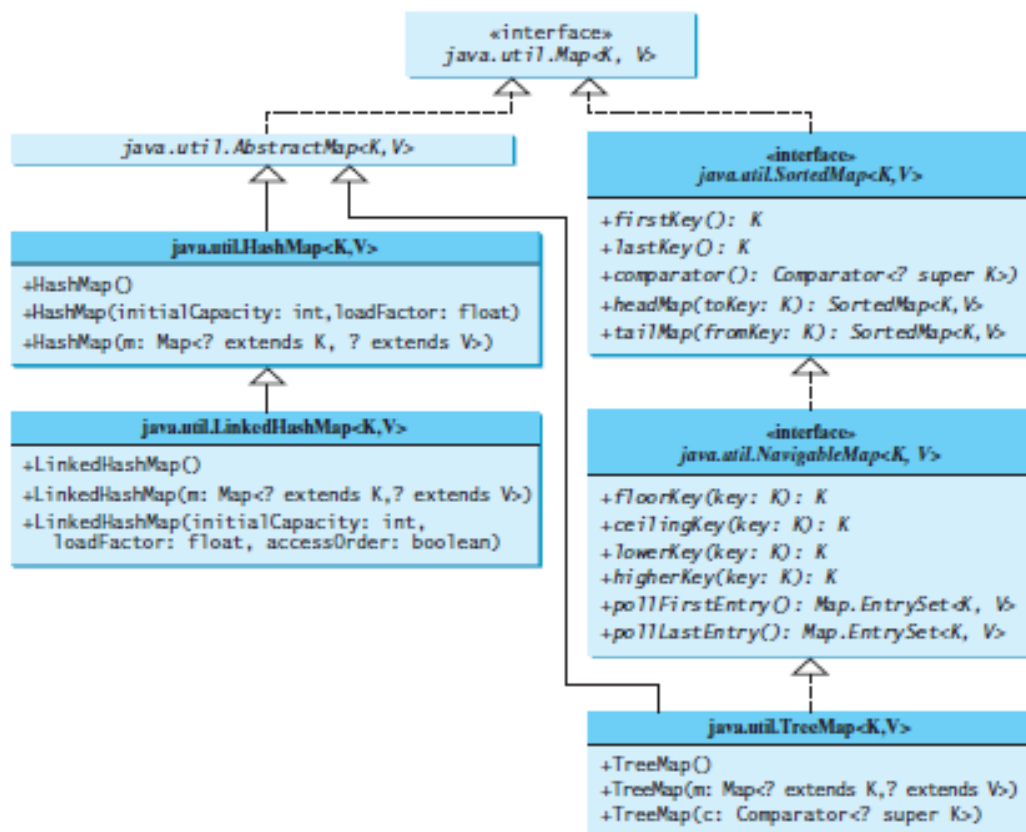
### Maps:

You can create a map using one of its three concrete classes: HashMap, LinkedHashMap, or TreeMap.

A map is a container object that stores a collection of key/value pairs. It enables fast retrieval, deletion, and updating of the pair through the key. A map stores the values along with the keys. The keys are like indexes. In List, the indexes are integers. In Map, the keys can be any objects. A map cannot contain duplicate keys. Each key maps to one value. A key and its corresponding value form an entry stored in a map, as shown in Figure a. Figure b shows a map in which each entry consists of a Social Security number as the key and a name as the value.



Maps[1]



Maps[1]

## Text and Binary I/O:

Java provides many classes for performing text I/O and binary I/O.

### Text I/O

Java provides many classes for performing text I/O and binary I/O.

Ex:

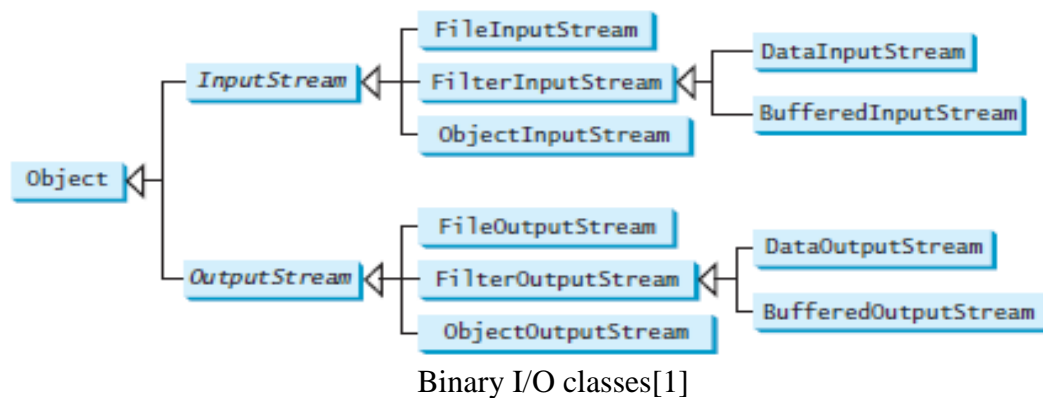
```

PrintWriter output = new PrintWriter("temp.txt");
output.print("Java 101"); //used to write in file.
output.close();
  
```

An input class contains the methods to read data, and an output class contains the methods to write data.

```
Scanner input = new Scanner(new File("temp.txt"));
System.out.println(input.nextLine());
```

### Binary I/O:



### Recursion:

Recursion is a technique that leads to elegant solutions to problems that are difficult to program using simple loops.

A recursive method is one that invokes itself. In some cases, recursion enables you to create an intuitive, straightforward, simple solution to a problem.

### Recursive Helper Methods:

Sometimes you can find a solution to the original problem by defining a recursive function to a problem similar to the original problem. This new method is called a recursive helper method. The original problem can be solved by invoking the recursive helper method.

### Programs:

**8.1 Demonstrate test time and removed time required for hash set, tree set, an array set, linked list.**

**Date:** //Write date of experiment here

**Procedure:**

//Write program here

**Observations:**

//Write program output here

**8.2 Write a Java program to find the longest word in a text file.**

**Date:** //Write date of experiment here

**Procedure:**

//Write program here

**Observations:**

//Write program output here

**8.3 Write a program to create a file name 123.txt, if it does not exist. Append a new data to it if it already exist. write 150 integers created randomly into the file using Text I/O. Integers are separated by space.**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion: //Write conclusion here.**

**8.4 Write a program that reads words from a text file and displays all the nonduplicate words in descending order. The text file is passed as a command-line argument .**

**Date: //Write date of experiment here**

**Procedure:**

**//Write program here**

**Observations:**

**//Write program output here**

**Conclusion: //Write conclusion here.**

**8.5 Write a program to find whether a String is palindrome or not by using recursive helper method.**

**Quiz:**

- 1. Differentiate Text I/o and Binary I/O**
- 2. Differentiate Recursion and Iteration.**
- 3. Describe Random access file.**

**Suggested Reference:**

- 1. Intro to Java Programming, 10<sup>th</sup> edition, Y.Daniel Liang, Pearson**
- 2. Complete reference Core JAVA by Herbert Schildt. MC Graw Hill.**

**References used by the students:**

**//Write your reference here**

**Rubric wise marks obtained:**

Rubrics	Knowledge (2)		Problem Recognition( 2)		Logic Building (2)		Completeness and accuracy (2)		Ethics (2)		Total
	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	
Marks											



## Practical- 9

### Aims:

Implement Following using Java FX

- 9.1 Write a program that moves a circle up, down, left or right using arrow keys.
- 9.2 Write a program that displays the color of a circle as red when the mouse button is pressed and as blue when the mouse button is released.
- 9.3 Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed.

**Relevant CO: CO5**

### Objectives:

To understand and implement javafx programs.

**Equipment/Instruments: jdk 11 , Any IDE.**

### Theory:

JavaFX is a Java library used to develop desktop applications with rich graphical user interfaces (GUIs). It was introduced as a replacement for Swing in Java 8 and is now the standard GUI library for Java.

JavaFX provides a wide range of GUI controls, including buttons, text fields, labels, menus, tables, and many others. It also includes support for multimedia and 3D graphics, making it a powerful platform for developing applications with advanced user interfaces.

One of the key features of JavaFX is its ability to use CSS to style the user interface. This makes it easy to customize the look and feel of the application to match specific branding or design requirements.

JavaFX can be used with a variety of development environments, including Eclipse, IntelliJ IDEA, and NetBeans. It is also supported by many third-party tools and libraries, making it a popular choice for developing desktop applications in Java.

The abstract `javafx.application.Application` class defines the essential framework for writing JavaFX programs.

For Ex:

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.stage.Stage;

public class MyJavaFX extends Application
{
    @Override // Override the start method in the Application class

    public void start(Stage primaryStage)
    {
```

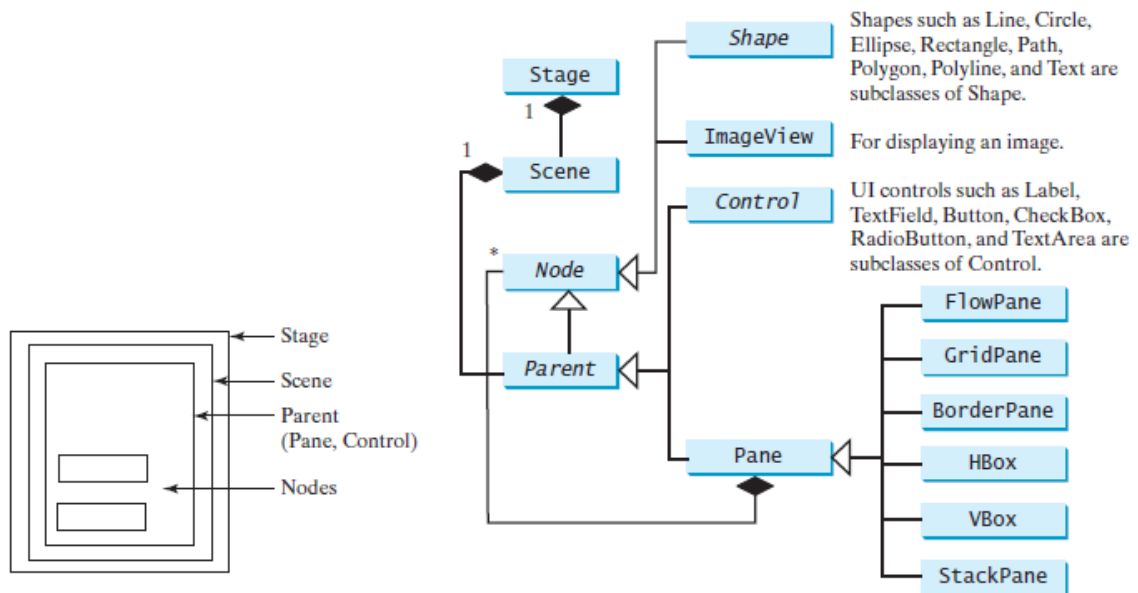
```

// Create a scene and place a button in the scene
Button btOK = new Button("OK");
Scene scene = new Scene(btOK, 200, 250);
primaryStage.setTitle("MyJavaFX"); // Set the stage title
primaryStage.setScene(scene); // Place the scene in the stage
primaryStage.show(); // Display the stage
}

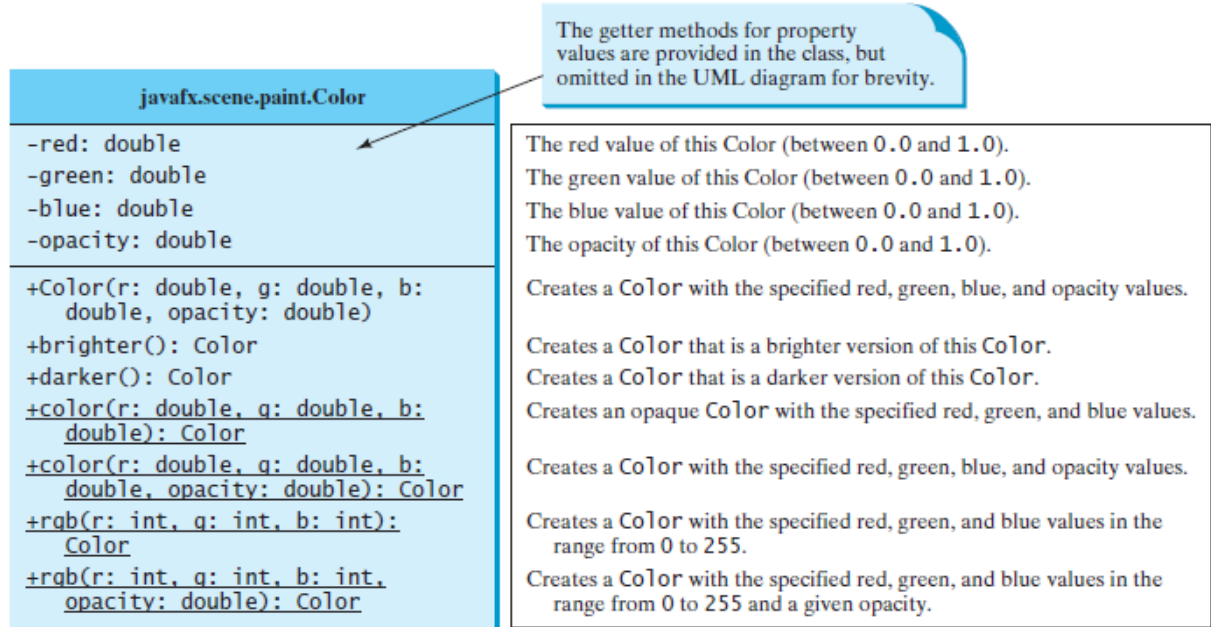
/**
 * The main method is only needed for the IDE with limited
 * JavaFX support. Not needed for running from the command line.
 */
public static void main(String[] args)
{
    Application.launch(args);
}
}

```

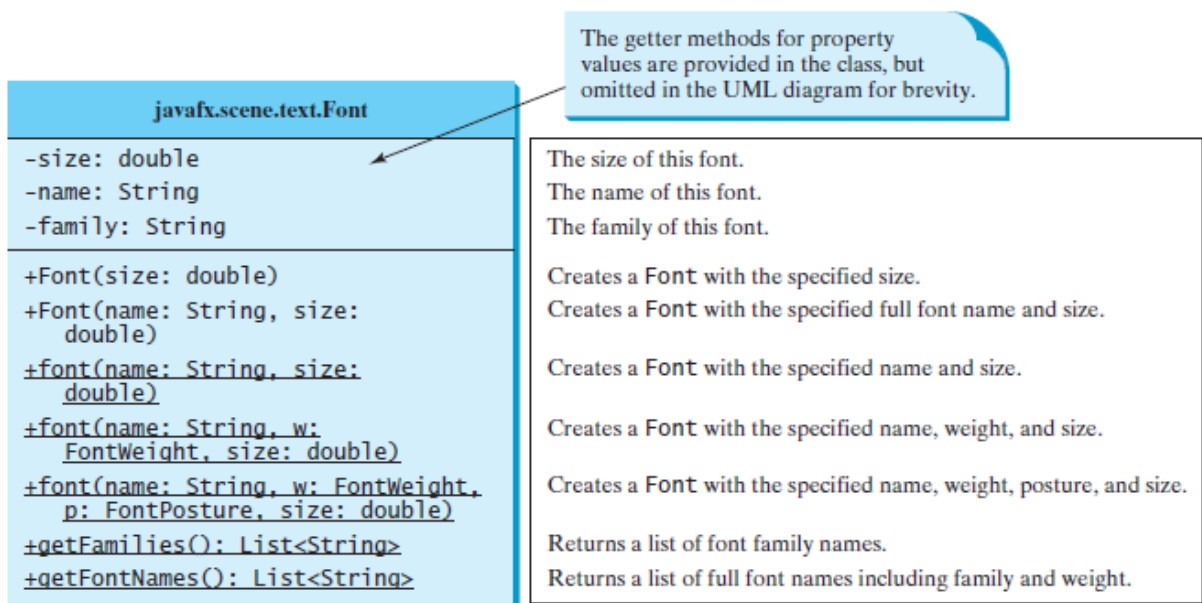
The launch method (line 22) is a static method defined in the Application class for launching a stand-alone JavaFX application. The main method (lines 21–23) is not needed if you run the program from the command line. It may be needed to launch a JavaFX program from an IDE with a limited JavaFX support. When you run a JavaFX application without a main method, JVM automatically invokes the launch method to run the application. The main class overrides the start method defined in `javafx.application.Application`. The start method normally places UI controls (in above example it is button) in a scene and displays the scene in a stage. A Stage object is a window.



Basic javafx class structure [1]

**Color Class:**

Color class[1]

**Font class:**

Font class[1]

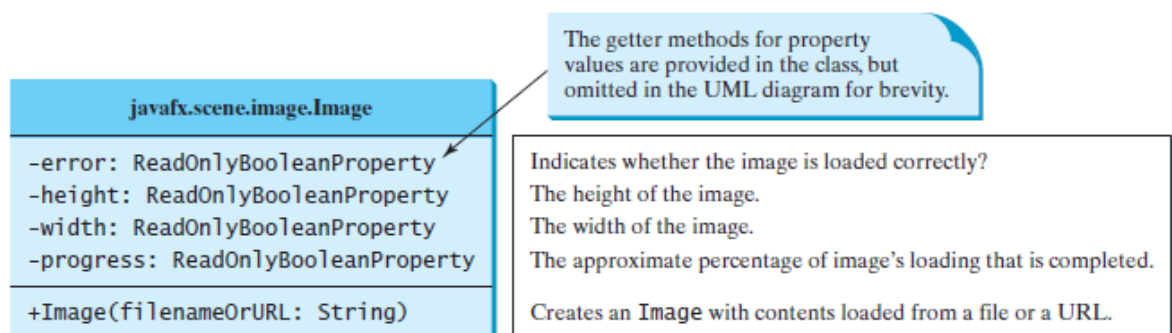
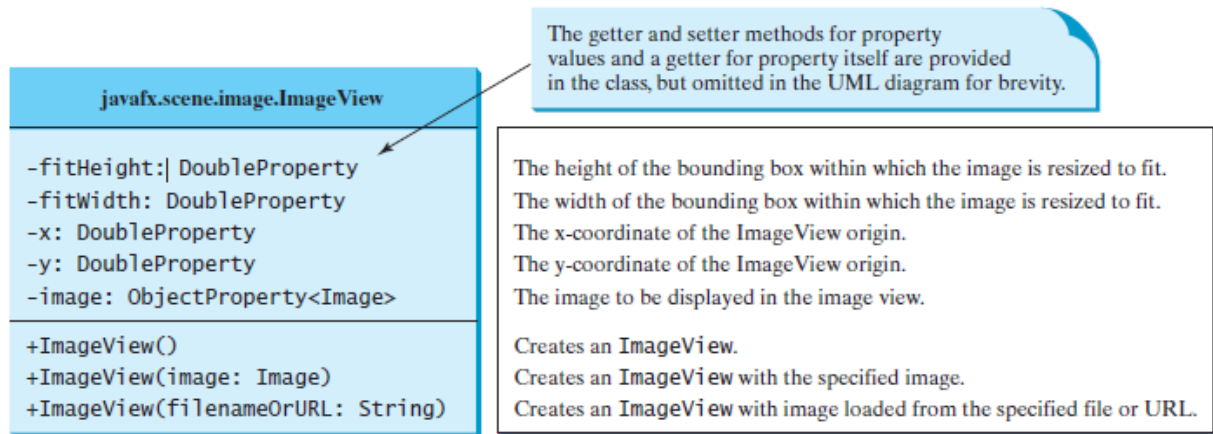
**Image and ImageView classes:**

Image Class[1]



ImageView class[1]

### Layout Panes:

JavaFX provides many types of panes for automatically laying out nodes in a desired location and size.

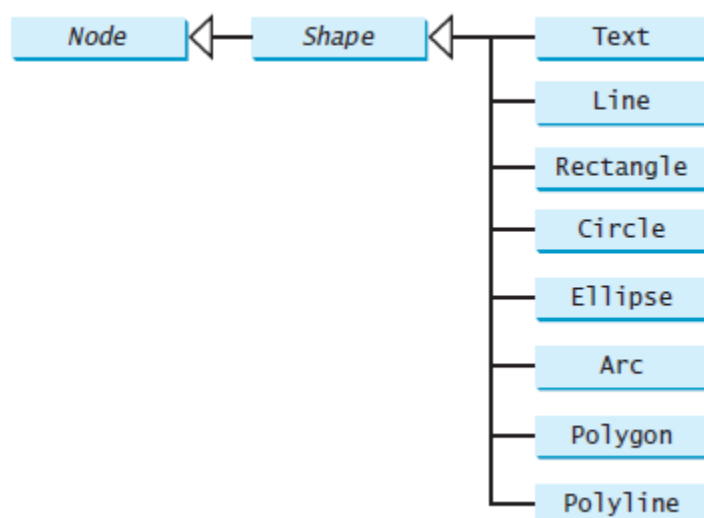
JavaFX provides many types of panes for organizing nodes in a container, as shown below [1].

Class	Description
<b>Pane</b>	Base class for layout panes. It contains the <code>getChildren()</code> method for returning a list of nodes in the pane.
<b>StackPane</b>	Places the nodes on top of each other in the center of the pane.
<b>FlowPane</b>	Places the nodes row-by-row horizontally or column-by-column vertically.
<b>GridPane</b>	Places the nodes in the cells in a two-dimensional grid.
<b>BorderPane</b>	Places the nodes in the top, right, bottom, left, and center regions.
<b>HBox</b>	Places the nodes in a single row.
<b>VBox</b>	Places the nodes in a single column.

### Shapes:

JavaFX provides many shape classes for drawing texts, lines, circles, rectangles, ellipses, arcs, polygons, and polylines.

The Shape class is the abstract base class that defines the common properties for all shapes.



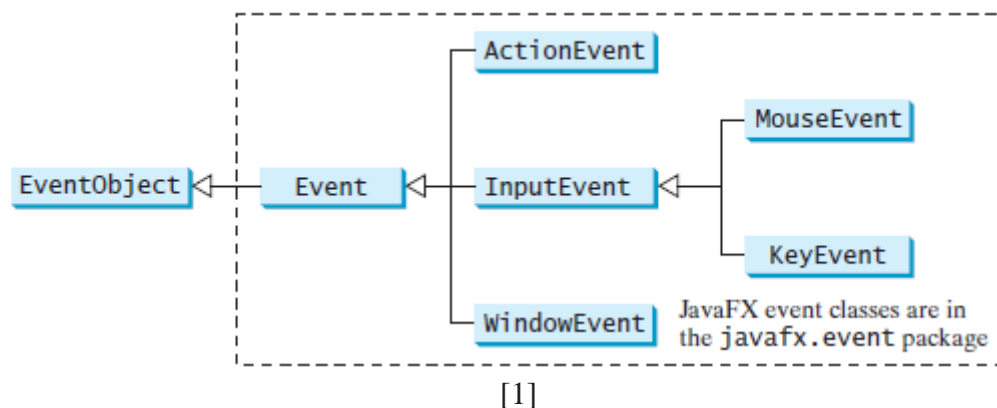
## Event Handling in javafx:

Event handling in JavaFX refers to the process of responding to user-generated events, such as mouse clicks, button presses, and key presses, in a JavaFX application.

In JavaFX, event handling involves the following steps:

1. **Registering an event handler:** An event handler is a piece of code that is executed in response to a specific event. To register an event handler in JavaFX, you typically use the `setOn<Event>()` method of a Node object, where `<Event>` is the type of event you want to handle (e.g. `setOnMouseClicked()` for handling mouse clicks).
2. **Implementing the event handling code:** Once you have registered an event handler, you need to implement the code that should be executed when the event occurs. This code typically resides in a method that takes an event object as its argument.
3. **Handling the event:** When the event occurs, the registered event handler is invoked, and the event object is passed to the method that implements the event handling code. The code in this method should then perform the necessary actions in response to the event.

Some common event types in JavaFX include `MouseEvent`, `KeyEvent`, `ActionEvent`, and `WindowEvent`, among others. Additionally, JavaFX provides a variety of convenience methods and classes for working with events, such as the `EventHandler` interface, which defines a standard interface for handling events in JavaFX.



An event object contains whatever properties are pertinent to the You can identify the source object of an event using the `getSource()` instance method in the `EventObject` class. The subclasses of `EventObject` deal with specific types of events, such as action events, window events, mouse events, and key events etc.

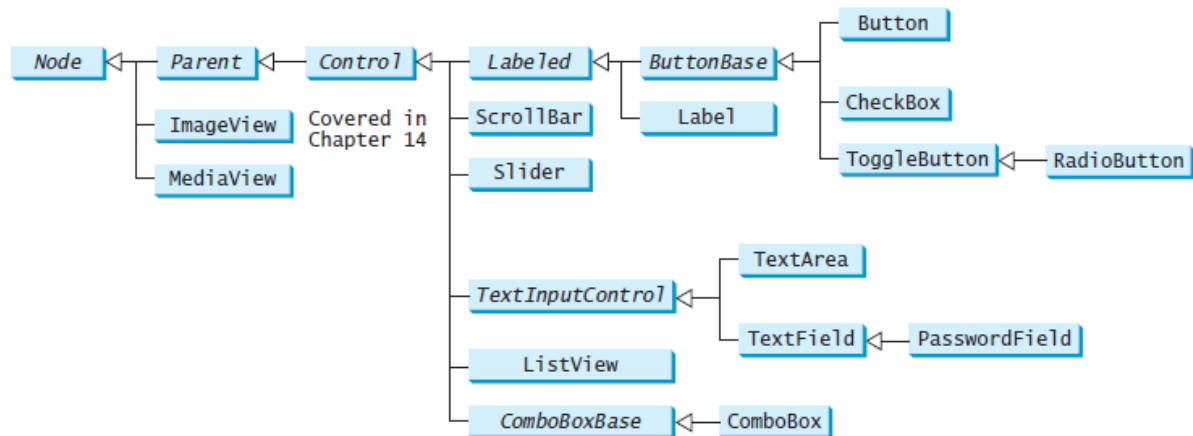
The handler object must be an instance of the corresponding event-handler interface to ensure that the handler has the correct method for processing the event. JavaFX defines a unified handler interface `EventHandler<T extends Event>` for an event `T`. The handler interface contains the `handle(T e)` method for processing the event. The handler object must be registered by the source object. Registration methods depend on the event type.

## Javafx UI control and MultiMedia:

JavaFX provides a wide range of UI controls and multimedia features that enable developers to create rich and interactive user interfaces for their applications.

Some of the key UI controls in JavaFX include:

1. Button: A control that represents a button that can be clicked by the user.
2. Label: A control that displays a piece of text.
3. TextField: A control that allows the user to enter text.
4. ComboBox: A control that displays a drop-down list of items that the user can select from.
5. ListView: A control that displays a list of items.
6. TableView: A control that displays a table of data.
7. WebView: A control that allows the user to display web content within a JavaFX application.



[1]

In addition to UI controls, JavaFX also provides support for multimedia features such as audio and video playback, as well as 2D and 3D graphics. Some of the key multimedia features in JavaFX include:

1. MediaPlayer: A class that provides support for audio and video playback.
2. MediaView: A control that allows the user to display audio and video content within a JavaFX application.
3. Image: A class that represents an image, which can be loaded from a file or created dynamically.
4. Canvas: A control that allows the user to draw 2D graphics within a JavaFX application.
5. 3D Graphics: JavaFX provides support for creating and rendering 3D graphics within a JavaFX application using the Java 3D API.

Overall, JavaFX provides a powerful set of tools for creating rich and interactive user interfaces that can incorporate multimedia features such as audio and video playback, as well as advanced 2D and 3D graphics.

## Programs:

**9.1 Write a program that moves a circle up, down, left or right using arrow keys.**

**Date: //Write date of experiment here**

## Procedure:

**//Write program here**

**Observations:****//Write program output here**

**9.2 Write a program that displays the color of a circle as red when the mouse button is pressed and as blue when the mouse button is released.**

**Date: //Write date of experiment here**

**Procedure:****//Write program here****Observations:****//Write program output here**

**9.3 Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed.**

**Date: //Write date of experiment here**

**Procedure:****//Write program here****Observations:****//Write program output here**

**Conclusion: //Write conclusion here.**

**Quiz:**

1. Describe Property binding in javafx.
2. Write advantages of javafx over AWT.
- 3.

**Suggested Reference:**

1. Intro to Java Programming, 10<sup>th</sup> edition, Y.Daniel Liang, Pearson
2. Complete reference Core JAVA by Herbert Schildt. MC Graw Hill.

**References used by the students:****//Write your reference here****Rubric wise marks obtained:**

Rubrics	Knowledge (2)		Problem Recognition(2)		Logic Building (2)		Completeness and accuracy (2)		Ethics (2)		Total
	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	
Marks											

## Practical- 10

**10. Develop an simple java application which uses Java FX.**

**Relevant CO: CO5**

**Objectives:**

To Develop small GUI based application.

**Equipment/Instruments: jdk 11 , Any IDE,**

**Theory:**

Using javafx Students have to develop one small application. The student must work in team. A team contains minimum 3 members and maximum 4 members.

All team member must have some predefined role in application development.

After successful implementation you have to create small report and power point presentation of your work.

Report contains following topics:

**Plagiarism Check certificate**( plagiarism must be less then 15%)

1. Introduction
2. Why you choose this application and how it is different from any existing app.
3. Roles of team members.
3. Working and Screen shots of application.
4. Conclusion and future scope
5. References

**Date:**

**//Write date of experiment here**

**Procedure:**

**//Place Report here**

**Rubric wise marks obtained:**

Rubrics	Knowledge (2)		Problem identification (2)		Team work (2)		Communication Skill (2)		Ethics (2)		Total
	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	
Marks											