Objectives: The objective of this lab assignment was to explore basic Java concepts, and implement classes and inheritance It aimed to understand the usage of pointers (references) and type casting in Java.

Learnings:

- **Basic Java Program:**
  - Created a basic Java program, "Welcome to JMI," within a class named "JavaAtJMI," compiled it to bytecode using the Java compiler, and executed the bytecode using the interpreter.
- **Java Setup:**
  - Checked the availability and version of the Java compiler on the machine.
  - Examined the historical context of Java's release and assessed its success as a technology based on its widespread usage, adaptability, and longevity.

- **Java Class Implementation and Inheritance:**
  - Implemented a **Point** class in Java with default and parameterized constructors, along with functions to calculate distance from another point (**distFrom**) and find the midpoint (**midPoint**).
  - Defined an **Element** class and extended it to the **Point** class, understanding the notion of inheritance in Java.
- **Point Cloud Operations and Type Casting:**
  - Generated a point cloud consisting of 10,000 random points and computed the centroid.
  - Explored the usage of Element type pointers, instanceOf operator, and type casting in Java to determine the object type and perform type conversions.

Challenges:

- Challenges to overcome with inheritance and references in Java, especially in implementing inheritance and type casting correctly.
- Calculating the centroid of a large set of points and manipulating object references required careful handling of data and object interactions.

Key Notes:

- Java, with its broad adoption, platform independence, and continual updates, can be considered a successful technology, offering portability, security, and scalability.
- In Java, there isn't a direct equivalent of the "virtual" keyword from C++; method overriding, and polymorphism are achieved without it.

Conclusion: This lab provide us the understanding of Java Programming, learning and implementation of inheritance, reference ,instantiating the object of inherited class and typecasting.

Lab Notes on JDK, Java SE and JRE

- **JDK (Java Development Kit):**
  - The JDK is a software development kit that includes tools necessary for Java application development.
  - It consists of the JRE (Java Runtime Environment), a Java compiler, Java Archive (JAR), debugger, and other development tools.
  - Developers use the JDK to create Java applications, compile, debug, and run them. It's a comprehensive package suitable for development purposes.
  - It includes libraries, APIs, and documentation to facilitate Java programming and development.
- **Java SE (Java Standard Edition):**
  - Java SE is a specification that defines the core Java platform aimed at desktop and server environments.
  - It provides the standard APIs, libraries, and tools for developing and running Java applications for general-purpose use.
  - Java SE includes basic libraries for tasks like input/output, networking, database access, and graphical user interface (GUI) development.
  - It forms the base edition for Java development, offering a standardized set of features across different platforms.
- **JRE (Java Runtime Environment):**
  - The JRE is an implementation of the Java Virtual Machine (JVM) that allows running Java applications on a specific platform.
  - It includes the Java class libraries, Java runtime interpreter, and other necessary runtime components required for executing Java programs.
  - Unlike the JDK, the JRE does not contain development tools or compilers. It is meant solely for running Java applications.