**Day 1**

**What is Java?**

 Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible.

**Advantages:**  
Platform Independence: Java's "write once, run anywhere" principle allows code to be executed on any device with a Java Virtual Machine (JVM), regardless of the underlying operating system or hardware. Strong Ecosystem: It boasts a vast ecosystem of libraries, frameworks (like Spring, Hibernate), and tools that aid in various types of development, making it easier for developers to build applications efficiently.  
**Disadvantages:**Performance: Java, while improved over the years, might not match the performance of languages like C or C++. The JVM's overhead and garbage collection can impact performance in certain scenarios. Memory Consumption: Java applications can consume a significant amount of memory due to the JVM's resource requirements, which might be a concern for resource-constrained environments.

**Java Components**

 Java Code(.java) – this is the java code you have written, and its saved in the .java file(s)  
Javac compiler – it compiles the java source code files (.java file) into bytecode so that it can be executed by JVM. The bytecode is saved in a class file by compiler.  
Bytecode - this is what is produced by the javac complier after compiling java code.  
Java Virtual Machine (JVM)  
Generally referred as JVM, it’s the primary function is to execute the bytecode produced by compiler.  
Each operating system has different JVM, however the output they produce after execution of bytecode is same across all operating systems – hence Java is referred regarded as platform independent.

**Main Java features**

Java main features includes but not limited to:  
Platform independent  
Object Oriented  
Simple  
Robust Language  
Secure  
Multithreading

**Day 2**

**What is a variable?**

In Java, a variable is a named memory location used to store data that can be changed during the program's execution. It's a fundamental concept in programming that allows you to store and manipulate values.

Java is a powerful and versatile programming language that has been a staple in the software development industry for several decades.

**Data types in Java**

 In java we have two categories of data type:  
1) Primitive data types, which represent simple, single values and are not objects.  
2) non-primitive data types, which refer to objects created from classes and are stored in memory as references or pointers to the actual objects. They specify the size and type of variable values, and it has no additional methods.

In Java, casting refers to the process of converting a value from one data type to another. There are two types of casting in primitive data types:  
Widening (Implicit) Casting: which occurs automatically when converting a smaller data type to a larger data type.  
Narrowing (Explicit) Casting: requires explicit specification and occurs when converting a larger data type to a smaller one.

**Operators in Java**

In Java, operators are special symbols or keywords used to perform operations on operands (variables, literals, or expressions). Java operators can be classified into: Arithmetic operators, which are used in mathematical expressions in the same way that they are used in algebra. Bitwise operators – works on bits and performs bit-by-bit operation. Logical Operators, Relational operators.

**Java Classes and Objects**

Classes and objects are fundamental concepts that form the basis of object-oriented programming (OOP). A class is a blueprint or template used to create objects. It defines the properties (attributes) and behaviours (methods) that objects of that class will possess. An object is an instance of a class. It is a real-world entity that has a unique state (attributes) and behaviour (methods) based on the class it belongs to.

**Day 3**

**What is OOP?**

OOP stands for Object-Oriented Programming. It's a programming paradigm that focuses on using objects and classes to design and build applications. OOP revolves around the concept of objects, which are instances of classes, and it emphasizes modularity, reusability, and extensibility in software development.

**Activity 1**

The popularity of object-oriented programming (OOP) over procedural and declarative programming can be attributed to several factors that align with the evolution of software development practices and industry needs:  
1. Modularity and Reusability  
2. Abstraction and Encapsulation  
3. Inheritance and Polymorphism

**Features of OOP – Core features**

Object-oriented programming (OOP) encompasses several core features that distinguish it from other programming paradigms. These features contribute to its effectiveness in modelling real-world scenarios and creating scalable and maintainable software solutions. The core features of OOP include:  
1. Abstraction  
2. Encapsulation  
3. Inheritance  
4. Polymorphism

**Features of OOP – Other features**

 Other features of OOP are following;  
Coupling, which refers to the knowledge or information or dependency of another class.  
Cohesion refers to the level of a component which performs a single well-defined task.  
Association represents the relationship between the objects.  
Aggregation is a narrower kind of association. It occurs when there’s a one-way (HAS-A) relationship between the two classes you associate through their objects.  
Composition is a stricter form of aggregation. It occurs when the two classes you associate are mutually dependent on each other and can’t exist without each other.

**Day 4**

**Introduction to Decision control**

Decision making structures have one or more conditions to be evaluated or tested by the program, along with a statement or statements that are to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

**If Statement, IF ELSE statement, Nested If Statement**

An if statement consists of a Boolean expression followed by one or more statements.  
Nested If Statement: an if statement can be followed by an optional else if...else statement, which is very useful to test various conditions using single if...else if statement.  
IF ELSE statement  
An if statement can be followed by an optional else statement, which executes when the Boolean expression is false.

**Activity 1**

A switch statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each case.  
The tenary operator only caters for two outcomes. Like the IF ELSE statement evaluates for the true or false only. It can only be used in place of if else statement.

**Day 4 Reflections**

In Java, decision control structures are used to make decisions based on certain conditions. These structures allow the program to execute different code blocks depending on whether a condition evaluates to true or false.

**Day 5**

**Java Identifiers**

Java Identifiers are the names of variables, methods, classes, packages and interfaces. Unlike literals they are not the things themselves, just ways of referring to them. In the HelloWorld program, HelloWorld, args, main and println are identifiers.

**Activity 1**

Access modifiers control the visibility of classes, methods, and attributes, while non-access modifiers modify the behaviour of classes, methods, and attributes in various ways, such as preventing inheritance, allowing constant values, or ensuring thread safety.

**Java Identifiers**

In Java, identifiers are names given to classes, methods, variables, packages, and other entities within a program. These names serve as labels to identify and reference these entities.

**Day 4 Reflections**

In Java, modifiers are keywords used to change the properties of classes, methods, variables, and other entities. They control access levels, behavior, and other characteristics of these elements within a Java program. Java modifiers can be categorized into several types:  
1. Access Modifiers  
2. Non-Access Modifiers