1. Object-oriented programming. Basic concepts: objects, inheritance, polymorphism, encapsulation.

Inheritance is making a sub class to get features from the parent class. You can use *extends*

**Polymorphism** means "many forms", and it occurs when we have many classes that are related to each other by inheritance.

Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit.

1. Class concept. Classes and objects in Java.

Classes and **Objects** are basic concepts of **Object Oriented Programming** which revolve around the real life entities.

1. Class members. Access modifiers.

The components of a **class**, such as its instance variables or methods are called the **members** of a **class** or **class members**.

The access modifiers in Java specifies the accessibility or scope of a field, method, constructor, or class.

1. Creation and initialization of objects. Calling methods.

An **initializer** is a line of code (or a block of code) placed outside any method, constructor, or other block of code.

To **call** a **method in Java**, write the **method's** name followed by two parentheses () and a semicolon; The process of **method calling** is simple. When a program invokes a **method**, the program control gets transferred to the called **method**.

1. Variable scopes.

**Java Scope**. In **Java**, **variables** are only accessible inside the region they are created. ...

Method **Scope**. **Variables** declared directly inside a method are available anywhere in the method following the line of code in which they were declared: ...

Block **Scope**. A block of code refers to all of the code between curly braces {}.

1. Modifiers finaland static.

When a **final modifier** is used with a class then the class cannot be extended further.

The *final* modifier for finalizing the implementations of classes, methods, and variables.

The *static* modifier for creating class methods and variables

A **static** inner **class** is a nested **class** which is a **static** member of the outer **class**. It can be accessed without instantiating the outer **class**, using other **static** members. Just like **static** members, a **static** nested **class** does not have access to the instance variables and methods of the outer **class**

1. Packages, instructions import.

A **java package** is a group of similar types of classes, interfaces and sub-packages.

Package in java can be categorized in two form, built-in package and user-defined package.

There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.