Unit – 05: Inheritance and package introduction and Exception Handling

- Inheritance basic,
- Method overriding,
- abstract class, Object class,
- Defining a package, access protection, importing a package,
- Introduction to interface, variables in interface, extension of interface,
- Fundamentals of Exception handling, types of exception, creating your own exception.
- Use of try and catch, nested try block, throw, throws, finally keywords.

Questions to be discussed:

1. What is inheritance in java? Explain its different types.
2. What do you mean by multi-level inheritance? Explain it with example.
3. Differentiate between method overloading and method overriding.
4. What do you mean by interface? Define it with example.
5. What do you mean by class modifier? Define it with example.
6. What is Exception in Java? Explain its type in brief.
7. What do you mean by un-checked exceptions? Explain it.
8. Explain various available Exception keywords in java.
9. Write short notes on:
   a. Abstract class
   b. Object class
   c. Try & catch keyword

CTH EDUCATION SAHARSA ALOK SIR

Inheritance in Java:

- Inheritance is an important concept of OOPs.
- It is a mechanism in which a child object acquires all the properties and behaviors of a parent object.
- Inheritance means driving a new class from an existing class.
- The existing class is known as **base class** or **super class** or **parent class**.
- The new class is known as a **derived class** or **sub class** or **child class**.
- Inheritance provides the **reusability** of code especially when there is a large scale of code to reuse.

Syntax:

```
Class A
{
        .........
        .........
}
class B extends A
{
        .........
        .........
}
```

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

Types of inheritance in java:

- There are four types of inheritance in java:
  1. Single Inheritance
  2. Multilevel Inheritance
  3. Hierarchical Inheritance
  4. Hybrid Inheritance

Single Inheritance:

- In single inheritance, a sub-class is derived from only one super class.
- It inherits the properties and behavior of a single-parent class.
- Sometimes it is also known as **simple inheritance**.

Single Inheritance

Multi-level Inheritance:

- In multi-level inheritance, a class is derived from a sub class which is also derived from another class.
- Note that the classes must be at different levels.
- It has a single base class and single derived class but multiple intermediate base classes.

Multi-level Inheritance

Hierarchical Inheritance:

- If a number of classes are derived from a single base class, it is called hierarchical inheritance.

Hierarchical Inheritance

Hybrid Inheritance:

- Hybrid means consist of more than one.
- Hybrid inheritance is the combination of two or more types of inheritance.

Hybrid Inheritance

Method Overriding in Java:

- If child class has the same method as declared in the parent class, it is known as method overriding.
- Method overriding is used for runtime polymorphism.
- Method overriding can't be achieved without inheritance.

Syntax:

```
Class A
{
        void run()
        {

        }
        Class B extends A
        {
                void run()
                {

                }
        }
}
```

Rules for Java Method Overriding

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

Difference between method overloading and method overriding in java:

Method Overloading	Method Overriding
Overloading is a compile-time polymorphism.	Overriding is a run-time polymorphism.
It helps to increase the readability of the program.	It is used to specific implementation of the method which is already provided by its parent class.
It occurs within the class.	It is performed in two classes.
Method overloading may or may not require inheritance.	Method overriding always needs inheritance.
In method overloading, methods must have the same name and different signatures.	In method overriding, methods must have the same name and same signature.
Poor Performance due to compile time polymorphism.	It gives better performance.

Abstract class in Java:

- Those class which has hiding the implementation and showing the function definition to the user is known as Abstract class.
- An abstract class must be declared with an abstract keyword.
- It can have abstract and non-abstract methods.
- We can't create an object of an abstract class.
- But we can create a reference variable of an abstract class because it refer to the objects of derived classes.

Object class in Java:

- The Object class is the parent class of all the classes in java by default.
- In other words, it is the topmost class of java.
- The Object class provides some common behaviors to all the objects such as object can be compared, object can be cloned, object can be notified etc.
- Object class is present in **java.lang** package.
- Every class in Java is directly or indirectly derived from the Object class.
- Therefore the Object class methods are available to all Java classes.
- Hence object class acts as a root of the inheritance hierarchy in any Java Program.

What is Package in Java?

- A java package is a collection of similar types of classes, interfaces and other packages.
- It works as containers for classes and other sub-packages.
- In other words we can say that the package is a way to group different classes or interfaces together.
- The grouping is done on the basis of functionality of classes and interfaces.
- There are two types of package in Java:
  1) Built-in Packages (packages from the Java API)
  2) User-defined Packages (create your own packages)

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

Import Package:

- We import the java package suing the keyword import.
- Suppose we want to use the student class stored in the java.util package then we can write the import statement at the beginning of the program like:

```
import.java.util.student
```

Access Protection in Packages:

- Packages adds another dimension to the access control.
- Java provides many levels of security that provides the visibility of members within the classes, subclasses, and packages.
- Access modifiers define the scope of the class and its members.
- In java, the accessibility of the members of a class or interface depends on its access specifiers.

There are four types of Java access modifiers:

1. Public
2. Private
3. Protected
4. Default

Private:

- The access level of a private modifier is only within the class.
- It cannot be accessed from outside the class.

Public:

- The access level of a public modifier is everywhere.
- It can be accessed from within the class, outside the class, within the package and outside the package.

Protected:

- The access level of a protected modifier is within the package and outside the package through child class.
- If you do not make the child class, it cannot be accessed from outside the package.

Default:

- The access level of a default modifier is only within the package.
- It cannot be accessed from outside the package.
- If you do not specify any access level, it will be the default.

Interface in Java:

- Interface is just like a class in java.
- An interface is a fully abstract class.
- It includes a group of abstract methods without a body.
- We use the interface keyword to create an interface in Java.
- Interface variables are by default public, static and final.
- Like abstract classes, we cannot create objects of interfaces.
- To use an interface, other classes must implement it.
- We use the implements keyword to implement an interface.

Extending an Interface:

- Similar to classes, interfaces can extend other interfaces.
- The extends keyword is used for extending interfaces.

  Example:

```
Interface Line
{
        ...............// members of Polygon interface
}
Interface Polygon extends Line
{
        ...............// members of Polygon interface
        ...............// members of Line interface
}
```

What do you mean by Exception in Java?

- Exception is an abnormal condition of the program execution.
- It is an unexpected event, which occurs during the execution of a program.
- When exception occur then it disrupts the normal flow of the program at run time.
- When an exception occurs within a method, it creates an object, this object is called the exception object.
- It contains information about the exception, such as the name and description of the exception and the state of the program when the exception occurred.

Major reasons why an exception Occurs:

- Invalid user input
- Device failure
- Loss of network connection
- Physical limitations (out of disk memory)
- Code errors

Exception Handling in Java:

- In Java, it is one of the powerful mechanism to handle the runtime errors so that the normal flow of the application can be maintained.

Types of Exception in Java?

There are two types of Exceptions:

1. **Checked Exceptions:**

- Checked exceptions are called compile-time exceptions because these exceptions are checked at compile-time by the compiler.

2. **Unchecked Exceptions:**

- The unchecked exceptions are just opposite to the checked exceptions.
- The compiler will not check these exceptions at compile time.
- In simple words, if a program throws an unchecked exception, and even if we didn't handle or declare it, the program would not give a compilation error.

Java Exception Keywords:

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

Keyword	Description
try	▪ The "try" keyword is used to specify a block where we should place an exception code. ▪ It means we can't use try block alone. ▪ The try block must be followed by either catch or finally.
catch	▪ The "catch" block is used to handle the exception. ▪ It must be preceded by try block which means we can't use catch block alone. ▪ It can be followed by finally block later.
finally	▪ The "finally" block is used to execute the necessary code of the program. ▪ It is executed whether an exception is handled or not.
throw	▪ The "throw" keyword is used to throw an exception.
throws	▪ The "throws" keyword is used to declare exceptions. ▪ It specifies that there may occur an exception in the method. ▪ It doesn't throw an exception. ▪ It is always used with method signature.