SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

23CSB101

OBJECT ORIENTED PROGRAMMING

UNIT II

INHERITANCE, PACKAGES AND INTERFACES

METHOD OVERRIDING

By

M.Kanchana

Assistant Professor/CSE

ABSTRACT CLASSES

.

An abstract class is a class that cannot be instantiated (we cannot create objects from it). It is used to define common behavior for multiple related classes while forcing them to implement their own versions of certain methods.

Abstract classes in Java enforce a "**must-do**" rule,

A teacher (abstract class) cannot exist without taking classes, but the way they teach might differ (lecture, discussion, practicals).

Different teachers (subclasses (English ,OOP) follow the rule of taking classes but in their own style.

ABSTRACT CLASSES

.

This concept is used **in online learning platforms** like Coursera, Udemy, or university portals where:

✓ School students take objective tests.

✓ College students take coding or written tests.

✓ The system enforces a "must-do" rule that all students must write exams but allows flexibility in the exam format.

ABSTRACT CLASSES

.

Properties of abstract class:

- **abstract** keyword is used to make a class abstract.
- Abstract class can’t be instantiated.
- If a class has abstract methods, then the class also needs to be made abstract using abstract keyword, else it will not compile.
- Abstract classes can have both concrete methods and abstract methods.
- The subclass of abstract class must implement all the abstract methods unless the subclass is also an abstract class.
- A constructor of an abstract class an be defined and can be invoked by the subclasses.
- We can run abstract class like any other class if it has main() method.

ABSTRACT CLASSES

```
abstract class Student {

    abstract void writeExam();
}
class SchoolStudent extends Student {
    @Override
    void writeExam() {
        System.out.println("School student is taking a multiple-choice exam.");
    }
}
class CollegeStudent extends Student {
    @Override
    void writeExam() {
        System.out.println("College student is taking a coding test.");
    }
}
```

ABSTRACT CLASSES

.

```
public class OnlineExamSystem {
    public static void main(String[] args) {
        Student student;


        student = new SchoolStudent();
        student.writeExam();


        student = new CollegeStudent();
        student.writeExam();
    }
}
```

final WITH INHERITANCE

Final is a keyword or reserved word in java used for restricting some functionality. It can be applied to member variables, methods, class and local variables in Java.
final keyword has three uses:

1. For declaring **variable** – to create a named **constant**. A final variable cannot be changed once it is initialized.

   final int MAXMARKS=100;

2. For declaring the **methods** – to prevent method overriding. A final method **cannot be overridden** by subclasses.

   final void run()

3. For declaring **the class** – to prevent a class from inheritance. A final class **cannot be inherited**.

   final class Bike{
   }

final WITH INHERITANCE-
Points to Remember:

1. A constructor cannot be declared as final.
2. Local final variable must be initializing during declaration.
3. All variables declared in an interface are by default final.
4. We cannot change the value of a final variable.
5. A final method cannot be overridden.
6. A final class cannot be inherited.
7. If method parameters are declared final then the value of these parameters cannot be changed.
8. It is a good practice to name final variable in all CAPS.
9. final, finally and finalize are three different terms. finally is used in exception handling
10. finalize is a method that is called by JVM during garbage collection.

THANK YOU