Ms. Nehal Adhvaryu

ABSTRACT CLASSES & METHODS & INTERFACE

Abstract Classes

- An abstract class is a class that is declared abstract. It may or may not include abstract methods.
- An Abstract class cannot be instantiated. it is only designed to be inherited.

Contd.

- A class containing one or more abstract methods implies that the class itself should be abstract.
- All abstract method must be defined with abstract keyword.

Syntax

- **accessModifier abstract class classname**

{

…field declared…

…abstract method…

…non-abstract method…

}

Example:

```
public abstract class Stud
{
        abstract void display();
}
```

Abstract Method

- An abstract method is treated as a pure virtual method.
- The methods which have only declaration and no method body in the class are called abstract method.
- An abstract method should be override in a subclass.

- Syntax:

accessModifier abstract returnType methodName();

Example:

Public abstract int setvalue();

Feature of Abstract class and Methods

- Static, private,& final methods cannot be abstract.These type of methods cannot be overridden.
- After declaring abstract method it must be defined in subclass.
- Any class with abstract method is automatically abstract itself and must be declared as such.

Interface

- Java does not support multiple inheritance, hence it provides an alternative abstraction known as Interface.
- Interface are specific reference data type and they share some common things with classes.
- Interfaces are syntactically similar to classes, but they lack instance variables, and their methods are declared without any body.
- Java interface should be implemented using keyword “implements”.

Interface

- When you implement an interface method, it must be declared as public. That means members of java interface are public by default.
- Interfaces are another means by which polymorphism is supported by java.
- Using interface, you can specify what a class must do, but not how it does it.

Difference between Interface and Class:-

INTERFACE	CLASS
Interface does not allow to create a object.	Class allows to create a object of it's associated type.
Interface are created to implement it's properties using class objects that implements it.	Class are made to create a object of It's associated type that uses Variable and methods included in class.
Variables declared inside of interface declaration is implicitly FINAL and STATIC, meaning they cannot be changed by the imple-menting class.	Variable declared inside of class can Be modified using methods of it's Class or methods of super class if Inherited.
There is no body in the methods Declared in interface.	Methods of class includes body part (may be Inline).

Difference between Interface and Class:-

INTERFACE

Using interface we can achieve the capabilities of multiple inheritance.

We can refer interface directly to Abstract class.

We can create interface using interface keyword.

```
Interface interface-name
{
Return-type method1(argument)
Return-type method2(argument)
}
```

CLASS

We cannot achieve multiple inheritance directly using classes.

To make the class abstract we need to use abstract keyword.

We can create class using class keyword.

```
Class class-name
{
Access-mode:
Return-type method1(argument)
Return-type method2(argument)
}
```

Difference between Interface and Abstract Class:-

INTERFACE	ABSTRACT CLASS
Variables declared inside of interface declaration is implicitly FINAL and STATIC, meaning they cannot be changed by the imple-menting class.	Variable declared inside of Abstract class can be modified using methods of it's sub class if Inherited.
There is no body in the methods Declared in interface.	Methods of Abstract class includes Body part(may be Inline).

Difference between Interface and Abstract Class:-

INTERFACE

Using interface we can achieve the capabilities of multiple inheritance.

We can create interface using interface keyword.

```
Interface interface-name
{
Return-type method1(argument)
Return-type method2(argument)
}
```

ABSTRACT CLASS

We cannot achieve multiple inheritance directly using abstract classes.

We can create abstract class using abstract keyword.

```
Abstract Class class-name
{
Access-mode:
Return-type method1(argument)
Return-type method2(argument)
}
```

Ms. Nehal Adhvaryu