Core Java

Lesson 06: Abstract Classes

Capgemini

Lesson Objectives

- After completing this lesson, participants will be able to:
  - Understand concept of Abstract classes
  - Extending Abstract class
  - Abstract class and Runtime polymorphism

Abstract Class

- Provides common behavior across a set of subclasses
- Not designed to have instances that work
- One or more methods are declared but may not be defined, these methods are abstract methods.
- Abstract method do not have implementation
- Advantages:
  - Code reusability
  - Help at places where implementation is not available

Abstract Class (cont..)

- Declare any class with even one method as abstract as *abstract*
- Cannot be instantiated
- Cannot use *Abstract* modifier for:
  - Constructors
  - Static methods
- Abstract class' subclasses should implement all methods or declare themselves as *abstract*
- Can have concrete methods also

Extending Abstract Class

- A class that is declared as abstract needs to be extended and its method implemented.
- It cannot be instantiated.

Extending Abstract Class

- Example of abstract class that has abstract method

```
abstract class Bike {
        abstract void run();
}

class Honda4 extends Bike {
        void run() {
                System.out.println("running safely..");
        }

        public static void main(String args[]) {
                Bike obj = new Honda4();
                obj.run();
        }
}
```

Demo

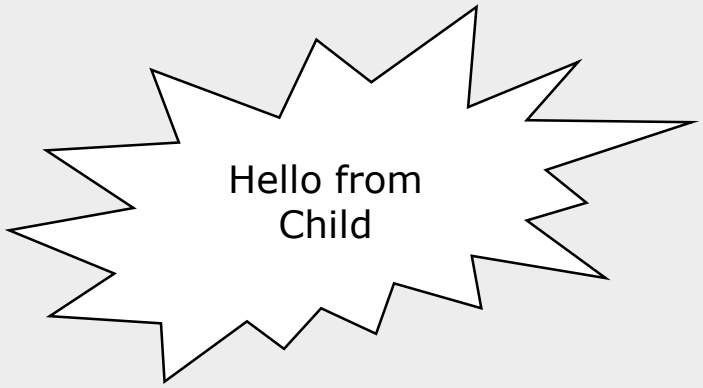
- Execute the Executor.java program

Runtime Polymorphism

- Runtime polymorphism enables a method can do different things based on the object used for invoking method at runtime
- Runtime polymorphism is implemented by doing method overriding

```
class Parent {
        public String sayHello() {
            return "Hello from
Parent";
        }
}
class Child extends Parent {
        public String sayHello() {
            return "Hello from
Child";
        }
}
```

```
Parent object = new
Child();
object.sayHello();
```



Demo

- Runtime polymorphism

Lab

- Lab 6: Abstract classes

Summary

- In this lesson, you have learnt about:
  - Abstract class
  - Extending abstract classes
  - Runtime Polymorphism

Review Question

- Question 1: Which of these is not a correct statement?
  - **Option 1:** Every class containing abstract method must be declared abstract
  - **Option 2:** Abstract class defines only the structure of the class not its implementation
  - **Option 3:** Abstract class can be initiated by new operator
  - **Option 4**: Abstract class can be inherited
- Question 2: Which of the following class definitions defines a legal abstract class?
  - **Option 1:**class A { abstract void unfinished() { } }
  - **Option 2:** class A { abstract void unfinished(); }
  - **Option 3:** abstract class A { abstract void unfinished(); }
  - **Option 4:** public class abstract A { abstract void unfinished(); }