Abstract Class & Abstract Methods - Notes

Abstract Class

Definition of Abstract Class:

A class which contains the 'abstract' keyword in its declaration is called an Abstract Class.

Key Points:

1. Abstract classes cannot be instantiated directly.
2. They may contain abstract methods (methods without a body) and non-abstract methods (with implementation).
3. Abstract classes can have constructors, fields, and methods like normal classes.
4. They are mainly used to provide a base for subclasses to extend and implement the abstract methods.
5. An abstract class may or may not have abstract methods.

Example in Java:

```
abstract class Animal {
    abstract void sound();
    void eat() {
        System.out.println("Animal is eating");
    }
}

class Dog extends Animal {
    void sound() {
        System.out.println("Dog barks");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal obj = new Dog();
```

```
        obj.sound();
        obj.eat();
    }
}
```

Abstract Methods:

Definition:

An abstract method is a method that is declared without implementation (without a body) inside an abstract class.

Key Points:

1. Abstract methods must be implemented in the first concrete subclass.
2. They are declared using the 'abstract' keyword.
3. They cannot have a body (implementation) in the abstract class itself.
4. All subclasses inheriting an abstract class must provide implementations for all abstract methods unless they are also declared abstract.

Example in Java:

```
abstract class Shape {
    abstract void draw();
}

class Circle extends Shape {
    void draw() {
        System.out.println("Drawing a Circle");
    }
}

class Rectangle extends Shape {
    void draw() {
```

```
        System.out.println("Drawing a Rectangle");
    }
}

public class Main {
    public static void main(String[] args) {
        Shape s1 = new Circle();
        Shape s2 = new Rectangle();
        s1.draw();
        s2.draw();
    }
}
```