Object Oriented Programming

Chapter 9

Ayesha Khatun

Lecturer, Green University of Bangladesh

Packages and Interfaces

Packages are containers for classes that are used to keep the class name space

Using the keyword **interface, you can fully abstract a class' interface from its implementation.**

Interfaces

- **interface keyword, Java allows you to fully utilize the "one interface,** multiple methods" aspect of polymorphism

```java
8
9  interface Callback {
10 void callback(int param);
11 }
12 class Client implements Callback {
13 // Implement Callback's interface
14 public void callback(int p) {
15 System.out.println("callback called with " + p);
16 }
17 void nonIfaceMeth() {
18 System.out.println("Classes that implement interfaces " +
19 "may also define other members, too.");
20 }
21 }
22 class Main {
23 public static void main(String args[]) {
24 Callback c = new Client();
25 c.callback(42);
26 }
27 }
```

Abstract class	Interface
1) Abstract class can **have abstract and non-abstract** methods.	Interface can have **only abstract** methods. Since Java 8, it can have **default and static methods** also.
2) Abstract class **doesn't support multiple inheritance.**	Interface **supports multiple inheritance.**
3) Abstract class **can have final, non-final, static and non-static variables.**	Interface has **only static and final variables.**
4) Abstract class **can provide the implementation of interface.**	Interface **can't provide the implementation of abstract class.**
5) The **abstract keyword** is used to declare abstract class.	The **interface keyword** is used to declare interface.
6) An **abstract class** can extend another Java class and implement multiple Java interfaces.	An **interface** can extend another Java interface only.
7) An **abstract class** can be extended using keyword "extends".	An **interface** can be implemented using keyword "implements".
8) A Java **abstract class** can have class members like private, protected, etc.	Members of a Java interface are public by default.

9)**Example:**
public abstract class Shape{
public abstract void draw();
}
	Example:
	public interface Drawable{
	void draw();
	}

Packages

	Private	No Modifier	Protected	Public
Same class	Yes	Yes	Yes	Yes
Same package subclass	No	Yes	Yes	Yes
Same package non-subclass	No	Yes	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes

Thank You!