



Romel Cabiling ▾



Home

Home > My courses > 121 - CC106 > MODULE 14: HOW TO WRITE JAVA PROGRAM USING ABSTRAC... > Lesson Proper for Week 14

Lesson Proper for Week 14

How to write Java Program using Abstraction and Interface?

An interface is a reference type in Java. It is similar to class. It is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface.

Along with abstract methods, an interface may also contain constants, default methods, static methods, and nested types. Method bodies exist only for default methods and static methods.

Writing an interface is similar to writing a class. But a class describes the attributes and behaviors of an object. And an interface contains behaviors that a class implements.

Unless the class that implements the interface is abstract, all the methods of the interface need to be defined in the class.

Advantages of interface in java:

Advantages of using interfaces are as follows:

Advantages of using interfaces are as follows:

1. Without bothering about the implementation part, we can achieve the security of implementation
2. In java, **multiple inheritance** is not allowed, however you can use interface to make use of it as you can implement more than one interface.

An interface is similar to a class in the following ways –

- An interface can contain any number of methods.



- An interface is written in a file with a **.java** extension, with the name of the interface matching the name of the file.
- The byte code of an interface appears in a **.class** file.
- Interfaces appear in packages, and their corresponding bytecode file must be in a directory structure that matches the package name.

However, an interface is different from a class in several ways, including –

- You cannot instantiate an interface.
- An interface does not contain any constructors.
- All of the methods in an interface are abstract.
- An interface cannot contain instance fields. The only fields that can appear in an interface must be declared both static and final.
- An interface is not extended by a class; it is implemented by a class.
- An interface can extend multiple interfaces.

Declaring Interfaces

The **interface** keyword is used to declare an interface. Here is a simple example to declare an interface –

Example

Following is an example of an interface –

```
/* File name : NameOfInterface.java */
import java.lang.*;
// Any number of import statements

public interface NameOfInterface {
    // Any number of final, static fields
    // Any number of abstract method declarations\
}
```

Interfaces have the following properties –

- An interface is implicitly abstract. You do not need to use the **abstract** keyword while declaring an interface.
- Each method in an interface is also implicitly abstract, so the abstract keyword is not needed.
- Methods in an interface are implicitly public.

Example

```
/* File name : Animal.java */
interface Animal {
    public void eat();
    public void travel();
}
```

Variables declared in interface are **public, static, and final** by default.



```

interface Try
{
    int a=10;
    public int a=10;
    public static final int a=10;
    final int a=10;
    static int a=0;
}

```

Differences between abstract class and interface that are given below.

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: <pre> public abstract class Shape{ public abstract void draw(); } </pre>	Example: <pre> public interface Drawable{ void draw(); } </pre>

Simply, abstract class achieves partial abstraction (0 to 100%) whereas interface achieves fully abstraction (100%).

Implementing Interfaces



When a class implements an interface, you can think of the class as signing a contract, agreeing to perform the specific behaviors of the interface. If a class does not perform all the behaviors of the interface, the class must declare itself as abstract.

A class uses the **implements** keyword to implement an interface. The implements keyword appears in the class declaration following the extends portion of the declaration.

Example

```
/* File name : MammalInt.java */
public class MammalInt implements Animal {
    public void eat() {
        System.out.println("Mammal eats");
    }
    public void travel() {
        System.out.println("Mammal travels");
    }
    public int noOfLegs() {
        return 0;
    }
    public static void main(String args[]) {
        MammalInt m = new MammalInt();
        m.eat();
        m.travel();
    }
}
```

This will produce the following result –

Output

Mammal eats

Mammal travels

When overriding methods defined in interfaces, there are several rules to be followed –

- Checked exceptions should not be declared on implementation methods other than the ones declared by the interface method or subclasses of those declared by the interface method.
- The signature of the interface method and the same return type or subtype should be maintained when overriding the methods.
- An implementation class itself can be abstract and if so, interface methods need not be implemented.

When implementing interfaces, there are several rules –

- A class can implement more than one interface at a time.
- A class can extend only one class, but implement many interfaces.
- An interface can extend another interface, in a similar way as a class can extend another class.



Extending Interfaces

An interface can extend another interface in the same way that a class can extend another class.

The **extends** keyword is used to extend an interface, and the child interface inherits the methods of the parent interface.

The following Sports interface is extended by Hockey and Football interfaces.

Example

```
// Filename: Sports.java
public interface Sports {
    public void setHomeTeam(String name);
    public void setVisitingTeam(String name);
}

// Filename: Football.java
public interface Football extends Sports {
    public void homeTeamScored(int points);
    public void visitingTeamScored(int points);
    public void endOfQuarter(int quarter);
}

// Filename: Hockey.java
public interface Hockey extends Sports {
    public void homeGoalScored();
    public void visitingGoalScored();
    public void endOfPeriod(int period);
    public void overtimePeriod(int ot);
}
```

The Hockey interface has four methods, but it inherits two from Sports; thus, a class that implements Hockey needs to implement all six methods. Similarly, a class that implements Football needs to define the three methods from Football and the two methods from Sports.

Extending Multiple Interfaces

A Java class can only extend one parent class. Multiple inheritance is not allowed. Interfaces are not classes, however, and an interface can extend more than one parent interface.

The extends keyword is used once, and the parent interfaces are declared in a comma-separated list.

For example, if the Hockey interface extended both Sports and Event, it would be declared as –

Example

```
public interface Hockey extends Sports, Events
```

EXERCISE 1

A student is to create a simple Java program about student enrolment system that will classify regular and irregular students. The student type is either 'R' for regular and 'I' for Irregular. What abstract method could be included in the *interface Enrolment*?



EXERCISE 2

Your school would like to create simple student enrolment system to help students enrolled in the college. Create an abstract method in the *interface Enrolment* and define the abstract method in the sub class, *class Enrolna implements Enrolment*

◀ Preliminary Activity for Week 14

Jump to...



Analysis, Application, and Exploration for Week 14 ▶



Navigation

Home

 Dashboard

Site pages

My courses

121 - CC106

Participants

 Grades

General

MODULE 1: WHAT IS APPLICATION DEVELOPMENT?

MODULE 2: WHAT ARE THE TECHNICAL SKILLS REQUIRED I...

MODULE 3: WHAT ARE THE PROGRAMMING LANGUAGES USED ...

MODULE 4: WHAT IS JAVA PROGRAMMING LANGUAGE AS APP...







MODULE 5: HOW TO WRITE JAVA PROGRAMMING LANGUAGE A...

MODULE 6: PRELIMINARY EXAMINATION

MODULE 7: HOW TO WRITE JAVA PROGRAM USING INTEGRAT...

MODULE 8: WHAT ARE THE BUILDING BLOCKS OF OBJECT-O...



MODULE 9: WHAT ARE THE BASIC CONCEPTS OF INHERITAN...
MODULE 10: WHAT ARE THE BASIC CONCEPTS OF ENCAPSUL...
MODULE 11: WHAT ARE THE BASIC CONCEPTS OF POLUMORP...
Week 12: Midterm Examination
MODULE 13: WHAT ARE THE BASIC CONCEPTS OF ABSTRACT...
MODULE 14: HOW TO WRITE JAVA PROGRAM USING ABSTRAC...
 Preliminary Activity for Week 14
 **Lesson Proper for Week 14**
 Analysis, Application, and Exploration for Week 14
 Generalization for Week 14
 Evaluation for Week 14
 Assignment for Week 14
MODULE 15: WHAT IS JAVA DATABASE CONNECTIVITY (JDB...
MODULE 16: WHAT ARE THE STEPS OF MANIPULATING DATA...
MODULE 17: EMERGING TECHNOLOGIES

121 - BPM101 / DM103
121 - OAELEC2
121 - ITE3
121 - MUL101
121 - ITSP2B
121 - WEB101 / CCS3218

Courses

Fair Warning

NOTICE: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for ***free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission.***

PROSECUTION: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following: Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense. Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.

COURSE OF ACTION: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.





visit www.bcp.edu.ph

Enrollment registration is now Ongoing

For 2nd Semester SY 2021 - 2022





We are accepting new students, returnees and transferees.

"Be trained to be the best,
Be linked to success"

 bcp-inquire@bcp.edu.ph  (8)442-8601 | (8)518-8050

The banner features a blue-tinted image of a multi-story building with a 'BCP' sign on the roof. A large white diagonal banner contains the enrollment information. The bottom right corner includes a quote and a circular logo with a graduation cap. The bottom of the banner has contact information for email and phone.

Activities

-  Assignments
-  Forums
-  Quizzes
-  Resources

Bestlink College of the Philippines
College Department

Powered by [eLearning Commons](#)

