|  |  |
| --- | --- |
| Course- BTech | Type- Core |
| Course Code- **CSET** | Course Name- **Object Oriented Programming Using Java** |
| Year- First | Semester- Even Batch- BTech 2nd Semester |

**Tutorial-9**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tutorial No.** | **Name** | **CO1** | **CO2** | **CO3** |
|  | **Interface** | -- |  | **--** |

**Objective:** The main objective of this tutorial is to learn about Interface of Java language.

* 1. What will be the Output of the below code:

interface A {  
 int methodA();  
}  
interface B {  
 int methodB();  
}  
interface C {  
 int methodC();  
}  
class D implements A, B, C {  
 int i = 999+111;

public int methodA() {  
 i =+ i / i;  
 return i;  
 }  
 public int methodB() {  
 i =- i \* i;  
 return i;  
 }  
 public int methodC() {  
 i = ++i - --i;  
 return i;  
 }  
}  
public class test {  
 public static void main(String[] args) {  
 D d = new D();  
 System.*out*.println(d.i);  
 System.*out*.println(d.methodA());  
 System.*out*.println(d.methodB());  
 System.*out*.println(d.methodC());  
 }  
}

**OUTPUT:**

**1110**

**1**

**-1**

**0**

* 1. What will be the Output of the below code: package doubt;

interface One {  
 int *i* = 222;  
 interface OneTwo {  
 int *i* = One.*i*+One.*i*;  
 interface OneTwoThree {  
 int *i* = OneTwo.*i* + OneTwo.*i*;  
 }  
 }  
}  
public class test {  
 public static void main(String[] args) {  
 System.*out*.println(One.OneTwo.OneTwoThree.*i*);  
 }  
}

**OUTPUT:**

**888**

* 1. What is the output of following code?

interface I {  
 class C {  
 int i;  
 public C(int i) {  
 this.i = ++i;  
 }  
 int methodC() {  
 return ++i;  
 }  
 }  
}  
public class test {  
 public static void main(String[] args) {  
 I.C c = new I.C(000);  
 System.*out*.println(c.methodC());  
 }  
}

**Output:**

**2**

* 1. Can we compile following code successfully?

interface I {  
 class C implements I {  
 public void methodI(int i) {  
 System.*out*.println(i);  
 }  
 }  
 void methodI(int i);  
}  
public class test {  
 public static void main(String[] args) {  
 I c = (I) new test();  
 }  
}

**OUTPUT:**

No, the provided code will not compile successfully due to several issues:

Nested Class C in Interface I:

An interface cannot contain a nested class with method implementations. Interfaces can only have nested interfaces or inner classes with constant variables.

The class C inside interface I is attempting to provide an implementation for the method methodI(int i), which is not allowed.

Method methodI(int i) in Interface I:

Interfaces can declare methods, but they cannot provide method implementations.

The method methodI(int i) inside interface I is missing a method body, which is not allowed in interfaces.

test Instantiation:

In the main method of test, the line I c = (I) new test(); attempts to instantiate test and assign it to a reference of type I.

However, test does not implement interface I, so this line will result in a compilation error.

To fix these issues, you need to redesign the structure of your code. If you intend to have a method methodI(int i) in interface I, you should declare it without a method body, and if you intend to have a nested class C, it should not contain method implementations. Additionally, if you want MainClass to implement I, you should ensure that MainClass provides implementations for all methods declared in interface I.

Correct –

interface I {  
 class C implements I {  
 public void methodI(int i) {  
 System.*out*.println("inner - "+i);  
 }  
 }  
 void methodI(int i);  
}  
public class test implements I {  
 public void methodI(int i) {  
 System.*out*.println("outer - "+i);  
 }  
 public static void main(String[] args) {  
 I c = (I) new test();  
 c.methodI(23);  
 C d = new C();  
 d.methodI(23);  
 }  
}

Output –

outer - 23

inner – 23

* 1. Is the following code written correctly?

class A

{

//Class A

}

interface B extends A

{

//Interface B extending Class A

}

Output:

No. An interface can extend another interface not the class.

* 1. What will be the output of the following program?

interface P {  
 String *p* = "PPPP";  
 String methodP();  
}  
interface Q extends P {  
 String *q* = "QQQQ";  
 String methodQ();  
}  
class R implements P, Q {  
 public String methodP() {  
 return *q*+*p*;  
 }  
 public String methodQ() {  
 return *p*+*q*;  
 }  
}  
public class test {  
 public static void main(String[] args) {  
 R r = new R();  
 System.*out*.println(r.methodP());  
 System.*out*.println(r.methodQ());  
 }  
}

**OUTPUT:**

**QQQQPPPP  
PPPPQQQQ**

* 1. Can you find out the errors in the following code?

interface A {  
 {  
 System.*out*.println("Interface A");  
 }  
  
 static {  
 System.*out*.println("Interface A");  
 }  
}

OUTPUT:  
Interfaces can’t have initializers.

* 1. How do you access interface field ‘i’ in the below code?

class P

{

interface Q

{

int i = 111;

}

}

OUTPUT:

P.Q.i

* 1. Can you identify the error in the below code?

interface A {  
 void methodA();  
}  
class B implements A {  
 public void methodA() {  
 interface C {  
 int *i* = 123;  
 void print\_i();  
 }  
 }  
}  
class test {  
 public static void main (String[] args){  
 B xyz = new B();  
 xyz.methodA();  
 System.*out*.println(xyz.i);  
 }  
}

OUTPUT:

Interfaces can’t be local members of a method.

* 1. What will be the output of the following program?

interface X {  
 char *c* = 'A';  
 char methodX();  
}  
class Y implements X {  
 {  
 System.*out*.println(*c*);  
 }  
 public char methodX() {  
 char c = this.*c*;  
 return ++c;  
 }  
}  
public class test {  
 public static void main(String[] args) {  
 Y y = new Y();  
 System.*out*.println(y.methodX());  
 System.*out*.println(y.*c*);  
 System.*out*.println(X.*c*);  
 }  
}

OUTPUT:

A  
B  
A  
A

Important links –

<https://www.geeksforgeeks.org/nested-classes-java/>

<https://www.geeksforgeeks.org/interface-nested-class-another-interface/>