

JAVA Assignment-4

Write a program to show Interface Example in java?

interface Bank

```
{
float rateOfInterest();
}
class SBI implements Bank
{
public float rateOfInterest()
{
return 9.15f;
}}
class PNB implements Bank
{
public float rateOfInterest()
{
return 9.7f;
}
}
class TestInterface2
{
public static void main(String[] args)
{
Bank b=new SBI();
System.out.println("ROI: "+b.rateOfInterest());
}
}
```

Write a program a Program with 2 concrete method and 2 abstract method in java.

```
abstract class AbstractExample
{
    abstract void display();
    // Concrete method
    void show()
    {
        System.out.println("Concrete method of abstract class");
    }
}

class SubClass extends AbstractExample
{
    void display()
    {
        System.out.println("Abstract method implemented");
    }
}

public class AbstractClass
{
    public static void main(String args[])
    {
        SubClass obj = new SubClass();
        obj.display();
        obj.show();
    }
}
```

Write a program to show the use of functional interface in java?

```
@FunctionalInterface
interface Square
{
    int calculate(int x);
}
class Test
{
    public static void main(String args[])
    {
        int a = 5;
        method Square s = (int x) -> x * x;
        int ans = s.calculate(a);
        System.out.println(ans);
    }
}
```

What is an interface in java?

The interface in Java is a mechanism to achieve abstraction. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java.

What is the use of interface in Java?

Use Java interface

It is used to achieve abstraction.

By interface, we can support the functionality of multiple inheritance.

It can be used to achieve loose coupling.

What is the lambda expression of Java 8?

A lambda expression is a short block of code which takes in parameters and returns a value. Lambda expressions are similar to methods

A lambda expression can be used to implement a functional interface without creating a class or an anonymous class. Lambda expressions can be used only with interfaces that declare a single method.

Syntax: (parameters...) -> { //Body }

Can you pass lambda expressions to a method? When?

A lambda expression passed in a method that has an argument of type of functional interface. If we need to pass a lambda expression as an argument, the type of parameter receiving the lambda expression argument must be of a functional interface type.

Example:

```
interface TestInterface
{
    boolean test(int a);
}

class Test
{
    static boolean check(TestInterface ti, int b)
    {
        return ti.test(b);
    }
}

public class LambdaExpressionPassMethodTest
{
    public static void main(String arg[])
    {
        boolean result = Test.check((x) -> (x%2) == 0, 10);
        System.out.println("The result is: "+ result);
    }
}
```

```
}  
}
```

What is the functional interface in Java 8?

An Interface that contains exactly one abstract method is known as functional interface.

What is the benefit of lambda expressions in Java 8?

- Lambda expressions improve code readability and do not require interpretation.
- Lambdas allow you to write concise code.
- It encourages the use of functional programming.
- It simplifies variable scope and encourages code reusability.
- Lambdas allow you to use parallel processing.
- Reduce the lines of code.

Is it mandatory for a lambda expression to have parameters?

No, it is not mandatory for a lambda expression to have parameters

The body of a lambda expression can contain zero, one, or more statements.

When there is a single statement curly brackets are not mandatory and the return type of the anonymous function is the same as that of the body expression.

When there is more than one statement, then these must be enclosed in curly brackets (a code block) and the return type of the anonymous function is the same as the type of the value returned within the code block, or void if nothing is returned.