**Assignment Questions 4**

💡 **Q1. Write a program to show Interface Example in java?**

interface Emotional {

    void expressLove();

    void expressFight();

}

class Person implements Emotional {

    public void expressLove() {

        System.out.println("I love you!");

    }

    public void expressFight() {

        System.out.println("Let's fight!");

    }

}

public class ConnectingClass {

    public static void main(String[] args) {

        Emotional person = new Person();

person.expressLove();

        person.expressFight();

    }

}

💡 **Q2.Write a program a Program with 2 concrete method and 2 abstract . method in java ?**

abstract class RussiaWar {

    // Abstract methods

    public abstract void declareWar();

    public abstract void endWar();

    // Concrete methods

    public void prepareTroops() {

        System.out.println("Preparing troops for the War");

    }

    public void celebrateVictory() {

        System.out.println("Celebrating victory in the War yeahhh…");

    }

}

class Country extends RussiaWar {

    public void declareWar() {

        System.out.println("declares war on its enemies.");

    }

    public void endWar() {

        System.out.println("The war come to an end.");

    }

    // Additional method specific future

    public void launchMissiles() {

        System.out.println("Launching nuclear missiles! (OH GOD)");

    }

}

public class Program {

    public static void main(String[] args) {

        RussiaWar ss = new Country();

        ss.declareWar();

        ss.prepareTroops();

        ss.launchMissiles();

        ss.endWar();

        ss.celebrateVictory();

    }

}

💡 **Q3.Write a program to show the use of functional interface in java?**

interface MyFunctionalInterface {

    void doSomething();

}

public class FunctionalInterfaceExample {

    public static void main(String[] args) {

MyFunctionalInterface myFunction = () -> System.out.println("Doing something!");

myFunction.doSomething(); } }

💡 **Q4.What is an interface in Java?**

In Java, an interface is a blueprint for a class that defines a set of method signatures. Classes implement interfaces to provide concrete implementations for those methods.

💡 **Q5.What is the use of interface in Java?**

1. Achieve abstraction and multiple inheritance.

2. Define a contract for classes that implement it.

3. Allow classes to be decoupled from their implementations.

4. Facilitate code reusability and maintainability.

💡 **Q6.What is the lambda expression of Java 8?**

Lambda expressions in Java 8 are concise and anonymous functions used to implement functional interfaces. They provide a simplified syntax for writing inline code. Key points:

Used with functional interfaces.

Shorter code with "->".

Supports both single-line and multi-line expressions.

Enables more readable and maintainable code for functional programming.

Facilitates the use of functional programming constructs like map, filter, reduce, etc.

💡 **Q7.Can you pass lambda expressions to a method? When?**

The method parameter is of a functional interface type.

To provide behavior as an argument to methods like sorting, filtering, mapping, etc.

To achieve custom actions/callbacks in event handling and concurrency.

Enables more flexible and concise code with reduced boilerplate.

💡 **Q8.What is the functional interface in Java 8?**

Functional interface in Java 8 is an interface with a single abstract method. It is used for lambda expressions and represents a single unit of behavior.

💡 **Q9.What is the benefit of lambda expressions in Java 8?**

The benefits of lambda expressions in Java 8 are concise and expressive code, improved readability, easy implementation of functional interfaces, and enhanced support for functional programming paradigms.

💡 [Q10. Is](http://Q10.Is) **it mandatory for a lambda expression to have parameters?**

 It is not mandatory for a lambda expression to have parameters in Java. A lambda expression can have zero, one, or multiple parameters, depending on the functional interface it implements. If the functional interface has no parameters, you can use an empty parameter list `()`. Lambda expressions provide concise syntax for implementing single abstract method interfaces.