Lab 9

Java Programs on Functional Interface, Lambda Expression, and Annotations

1. WAP to implement Functional Interface

```
@FunctionalInterface
interface MyFunctionalInterface {
   void display();
}

public class FunctionalInterfaceExample {
   public static void main(String[] args) {
      MyFunctionalInterface obj = () -> System.out.println("Hello from Functional
Interface!");
      obj.display();
   }
}
```

2. WAP to implement Lambda Expression

```
@FunctionalInterface
interface Addition {
   int add(int a, int b);
}

public class LambdaExample {
   public static void main(String[] args) {
      Addition sum = (a, b) -> a + b;
      System.out.println("Sum: " + sum.add(10, 20));
   }
}
```

3. Write a Java program to implement a lambda expression to check if a given string is empty.

```
@FunctionalInterface
```

```
interface StringCheck {
   boolean isEmpty(String s);
}

public class CheckEmptyString {
   public static void main(String[] args) {
      StringCheck check = (s) -> s.isEmpty();

      String str1 = "";
      String str2 = "Hello";

      System.out.println("Is str1 empty? " + check.isEmpty(str1));
      System.out.println("Is str2 empty? " + check.isEmpty(str2));
   }
}
```

4. WAP for Method References

```
interface Sayable {
   void say();
}

public class MethodReferenceExample {
   public static void saySomething() {
      System.out.println("Hello, this is a static method reference!");
   }

   public static void main(String[] args) {
      // Reference to a static method
      Sayable say1 = MethodReferenceExample::saySomething;
      say1.say();

      // Reference to an instance method
      MethodReferenceExample obj = new MethodReferenceExample();
      Sayable say2 = obj::instanceMethod;
      say2.say();
   }

   public void instanceMethod() {
      System.out.println("Hello, this is an instance method reference!");
   }
```

```
}
```

5. WAP for Default and Static Method in Interface Inner Class

```
interface MyInterface {
   void display();

   default void defaultMethod() {
      System.out.println("This is a default method in interface");
   }

   static void staticMethod() {
      System.out.println("This is a static method in interface");
   }

   class InnerClass {
      void show() {
         System.out.println("This is an inner class inside an interface");
      }
   }
}

public class DefaultStaticInnerExample implements MyInterface {
   public void display() {
      System.out.println("Implementing abstract method of interface");
   }

   public static void main(String[] args) {
      DefaultStaticInnerExample obj = new DefaultStaticInnerExample();
      obj.display();
      obj.defaultMethod();
      MyInterface.staticMethod();

      MyInterface.InnerClass inner = new MyInterface.InnerClass();
      inner.show();
   }
}
```

6. WAP to implement different types of Annotations in JAVA

```
class Parent {
   void show() {
```

```
        System.out.println("Parent class method");
    }
}

class Child extends Parent {
    @Override
    void show() {
        System.out.println("Child class overridden method");
    }

    @Deprecated
    void oldMethod() {
        System.out.println("This method is deprecated");
    }
}

public class AnnotationExample {
    @SuppressWarnings("deprecation")
    public static void main(String[] args) {
        Child obj = new Child();
        obj.show();
        obj.oldMethod(); // Deprecated method
    }
}
```