Ibraheem

42896

Bscs 5

2nd last lab

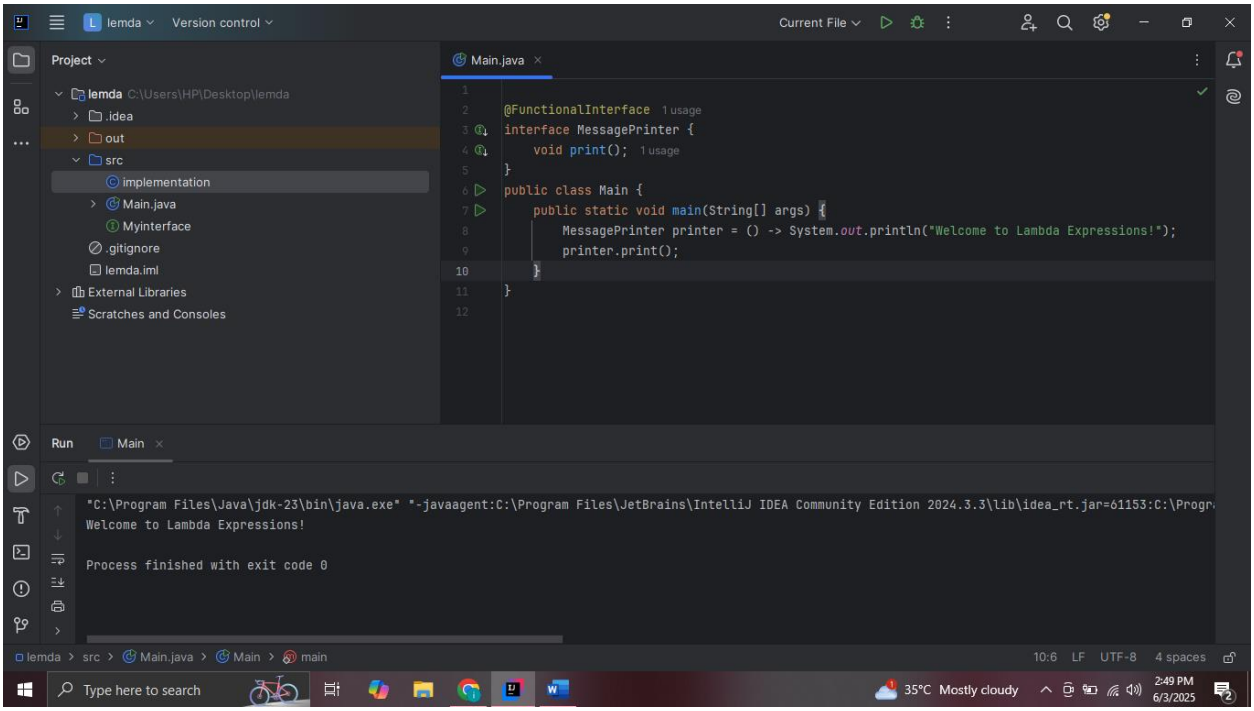
lambda

Task 1 :

Code :

```
@FunctionalInterface
interface MessagePrinter {
    void print();
}
public class Main {
    public static void main(String[] args) {
        MessagePrinter printer = () -> System.out.println("Welcome to Lambda
Expressions!");
        printer.print();
    }
}
```

Output :

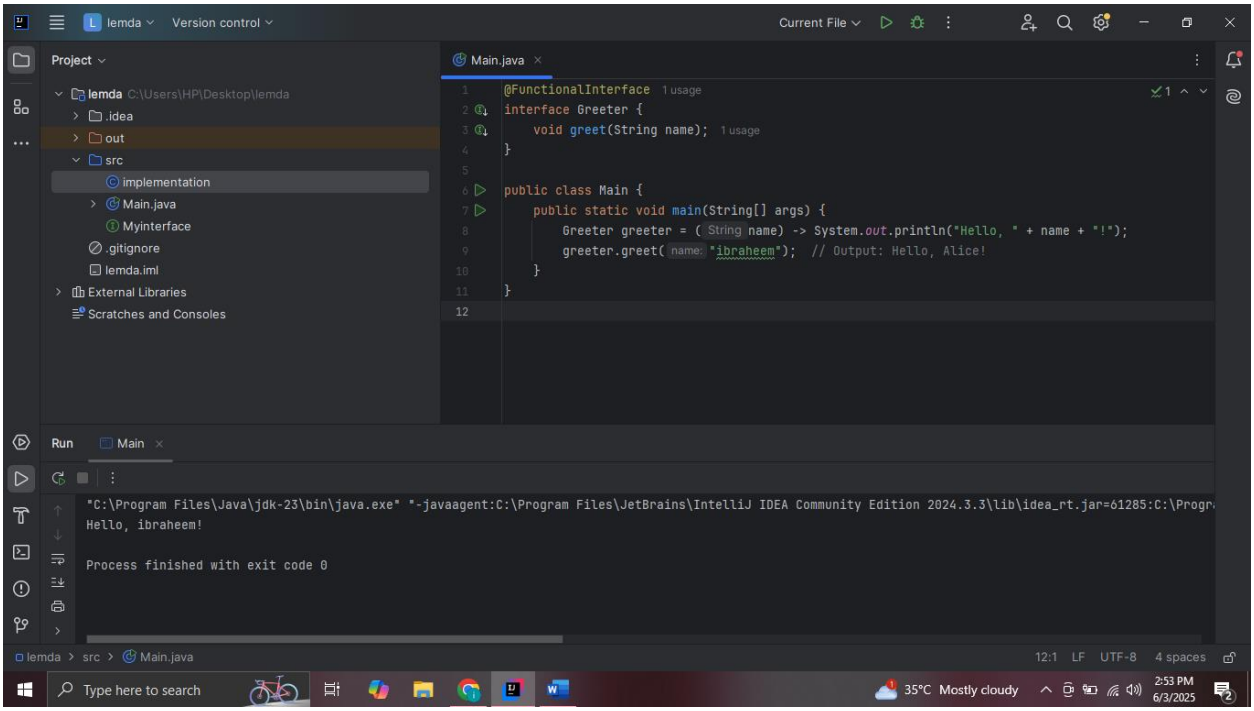


Task 2 :

```
@FunctionalInterface
interface Greeter {
    void greet(String name);
}

public class Main {
    public static void main(String[] args) {
        Greeter greeter = (name) -> System.out.println("Hello, " + name +
"!");
        greeter.greet("ibraheem");  // Output: Hello, Alice!
    }
}
```

Output :



Task 3 :

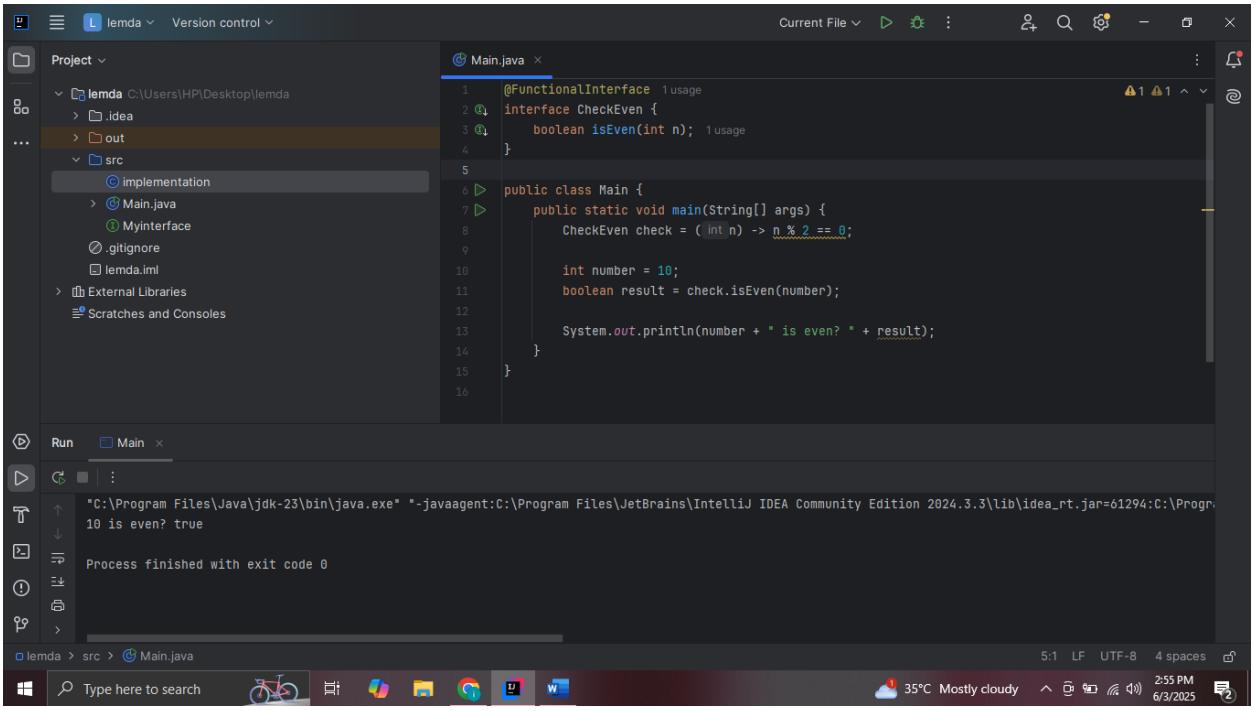
```java
@FunctionalInterface
interface CheckEven {
    boolean isEven(int n);
}

public class Main {
    public static void main(String[] args) {
        CheckEven check = (n) -> n % 2 == 0;

        int number = 10;
        boolean result = check.isEven(number);

        System.out.println(number + " is even? " + result);
    }
}
```

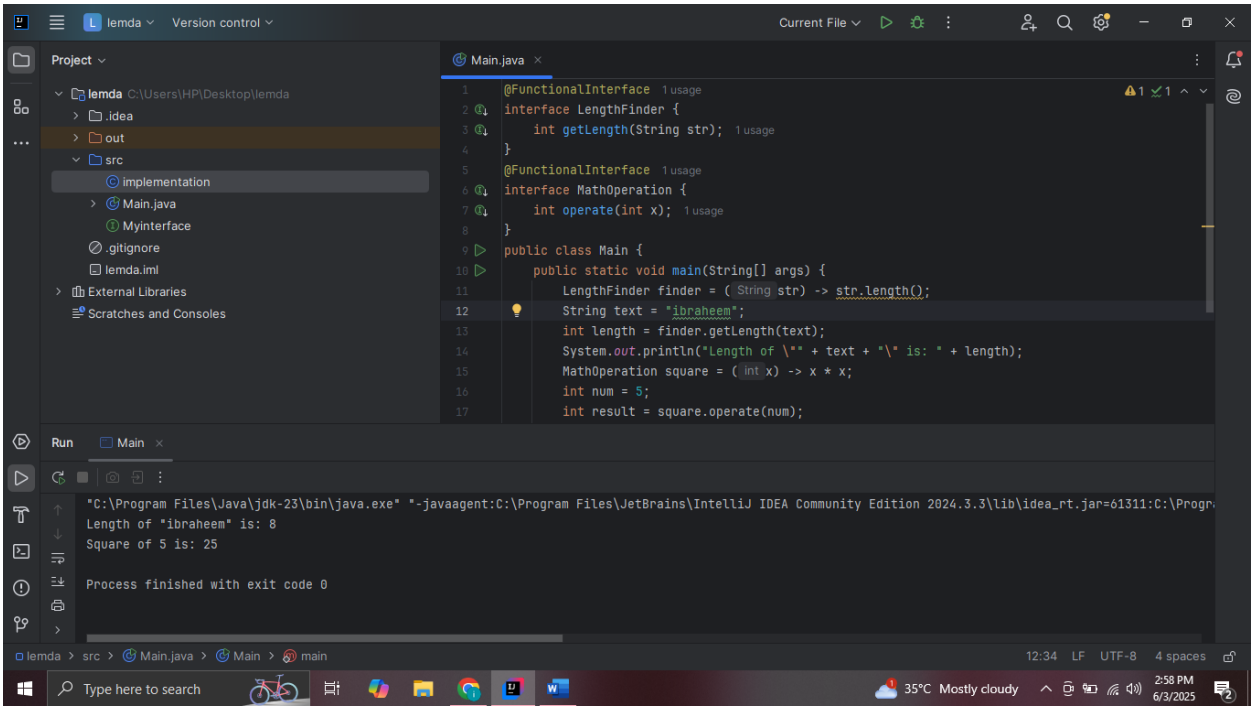
Output :



Task 4 :

```java
@FunctionalInterface
interface LengthFinder {
    int getLength(String str);
}
@FunctionalInterface
interface MathOperation {
    int operate(int x);
}
public class Main {
    public static void main(String[] args) {
        LengthFinder finder = (str) -> str.length();
        String text = "ibraheem";
        int length = finder.getLength(text);
        System.out.println("Length of \"" + text + "\" is: " + length);
        MathOperation square = (x) -> x * x;
        int num = 5;
        int result = square.operate(num);
        System.out.println("Square of " + num + " is: " + result);
    }
}
```

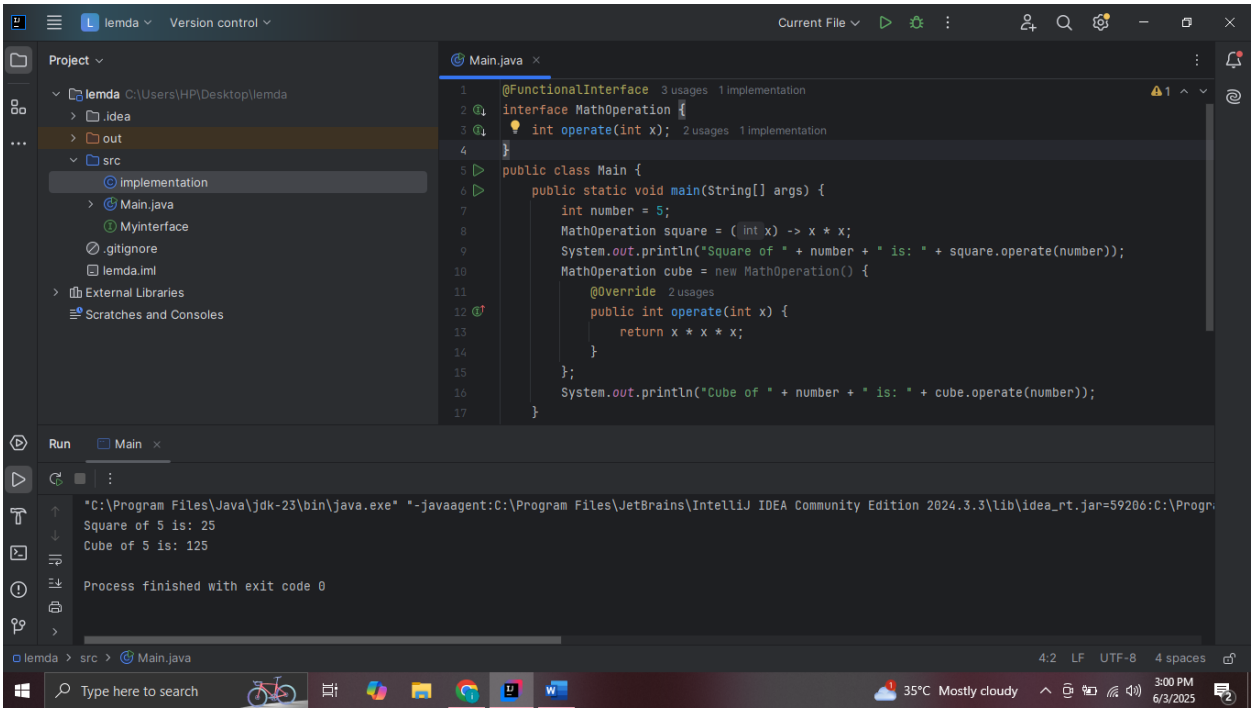
Output :



Task 5 :

```
@FunctionalInterface
interface MathOperation {
    int operate(int x);
}
public class Main {
    public static void main(String[] args) {
        int number = 5;
        MathOperation square = (x) -> x * x;
        System.out.println("Square of " + number + " is: " +
square.operate(number));
        MathOperation cube = new MathOperation() {
            @Override
            public int operate(int x) {
                return x * x * x;
            }
        };
        System.out.println("Cube of " + number + " is: " +
cube.operate(number));
    }
}
```

Output :

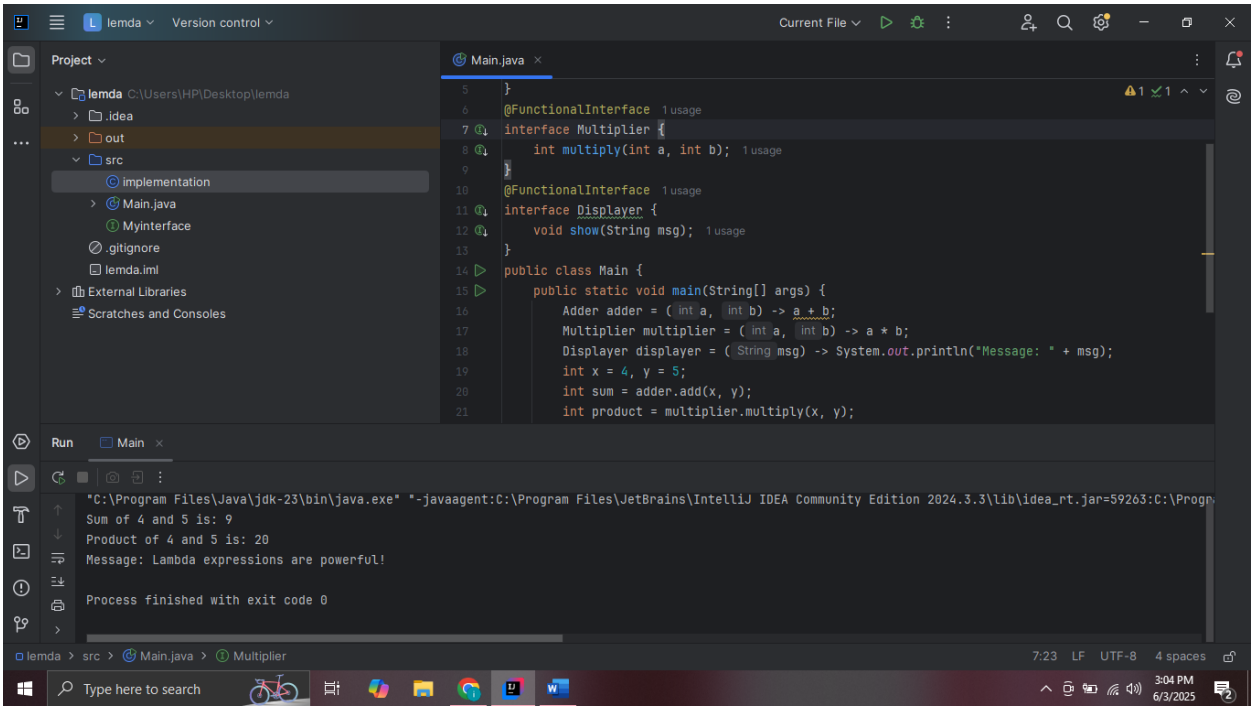


Task 6 :

```
@FunctionalInterface
interface Adder {
    int add(int a, int b);
}
@FunctionalInterface
interface Multiplier {
    int multiply(int a, int b);
}
@FunctionalInterface
interface Displayer {
    void show(String msg);
}
public class Main {
    public static void main(String[] args) {
        Adder adder = (a, b) -> a + b;
        Multiplier multiplier = (a, b) -> a * b;
        Displayer displayer = (msg) -> System.out.println("Message: " + msg);
        int x = 4, y = 5;
        int sum = adder.add(x, y);
        int product = multiplier.multiply(x, y);
        System.out.println("Sum of " + x + " and " + y + " is: " + sum);
        System.out.println("Product of " + x + " and " + y + " is: " +
product);
        displayer.show("Lambda expressions are powerful!");
```

```
        }
}
```

Output :


lemda
Version control
Current File
Project
lemda C:\Users\HP\Desktop\lemda
.idea
out
src
implementation
Main.java
Myinterface
.gitignore
lemda.iml
External Libraries
Scratches and Consoles
Main.java
@FunctionalInterface 1 usage
interface Multiplier {
int multiply(int a, int b); 1 usage
@FunctionalInterface 1 usage
interface Displayer {
void show(String msg); 1 usage
public class Main {
public static void main(String[] args) {
Adder adder = (int a, int b) -> a + b;
Multiplier multiplier = (int a, int b) -> a * b;
Displayer displayer = (String msg) -> System.out.println("Message: " + msg);
int x = 4, y = 5;
int sum = adder.add(x, y);
int product = multiplier.multiply(x, y);
Run
Main
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.3\lib\idea_rt.jar=59263:C:\Progr
Sum of 4 and 5 is: 9
Product of 4 and 5 is: 20
Message: Lambda expressions are powerful!
Process finished with exit code 0
lemda > src > Main.java > Multiplier
7:23 LF UTF-8 4 spaces
Type here to search
3:04 PM
6/3/2025