

Ibraheem

42896

Bscs 5

2<sup>nd</sup> last lab

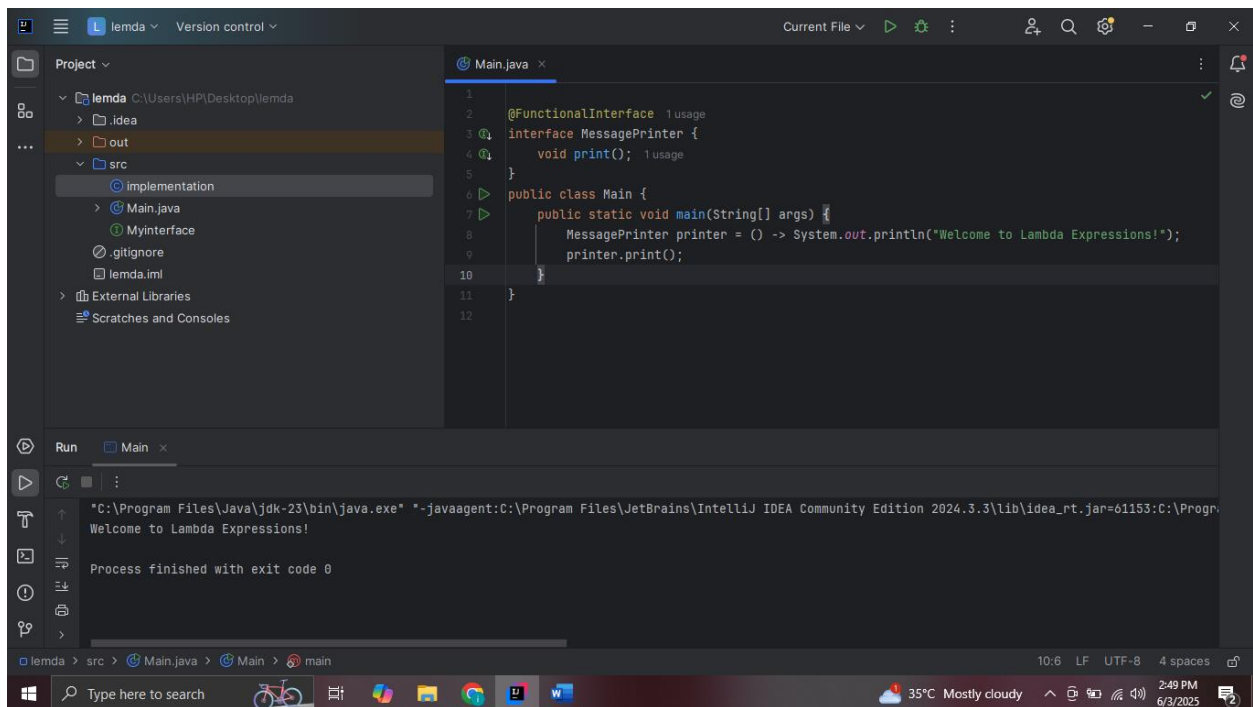
lambda

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 :

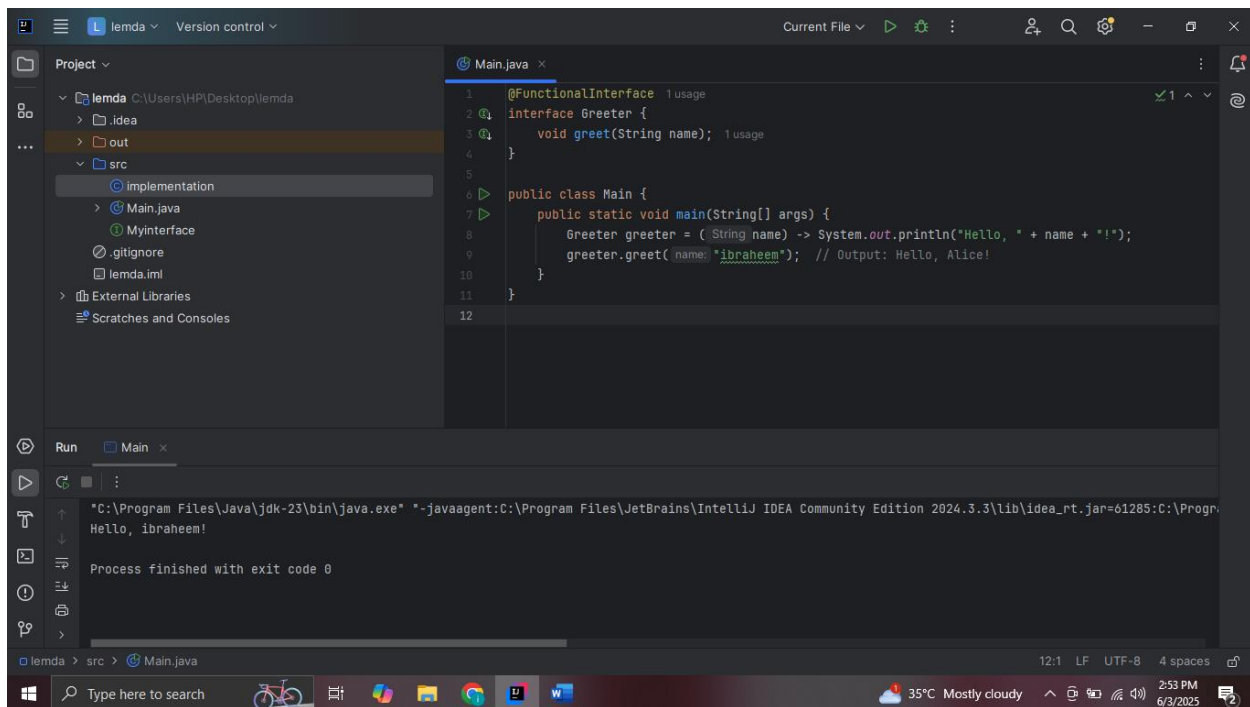


## Task 2 :

```
@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 :

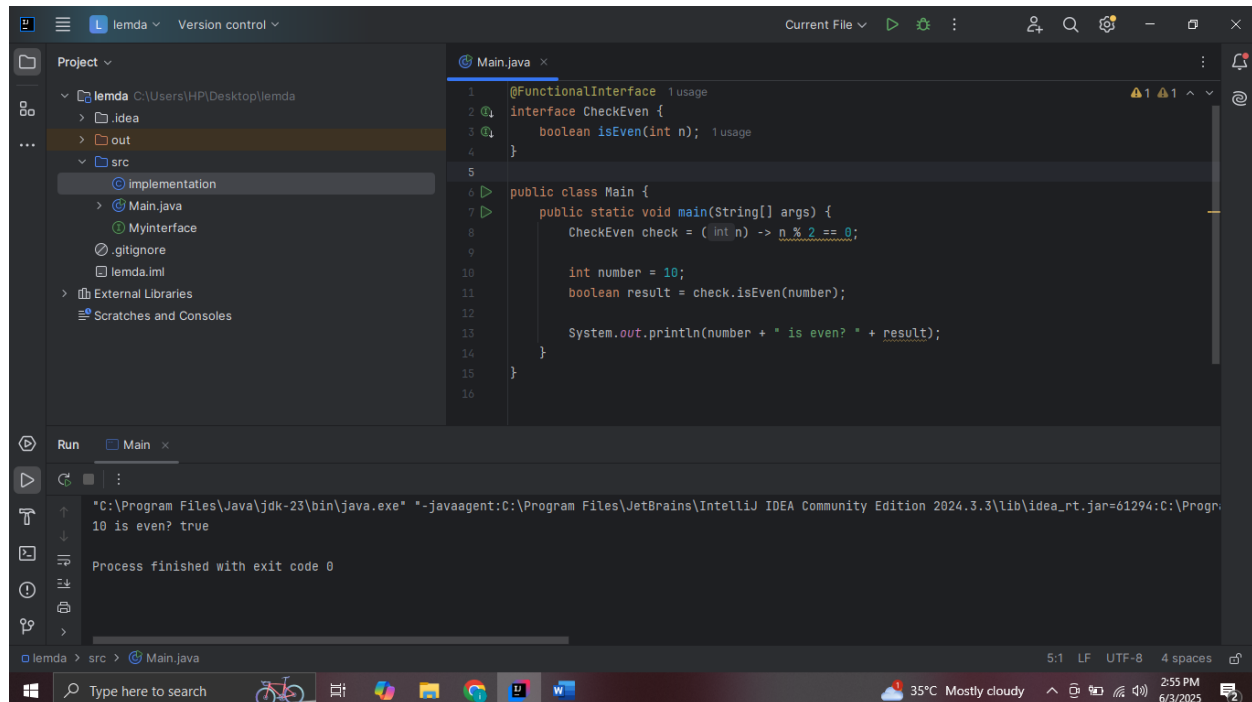
```
@FunctionalInterface
interface CheckEven {
    boolean isEven(int n);
}

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);
    }
}
```

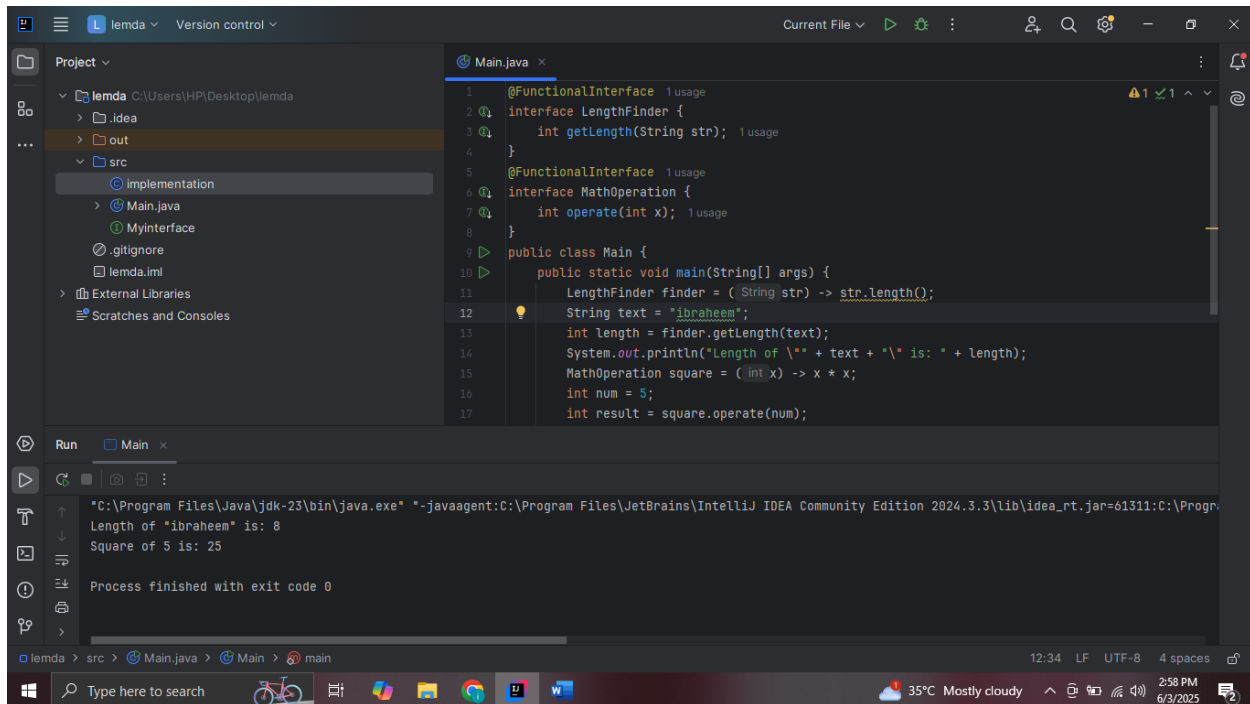
## Output :



## Task 4 :

```
@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 :

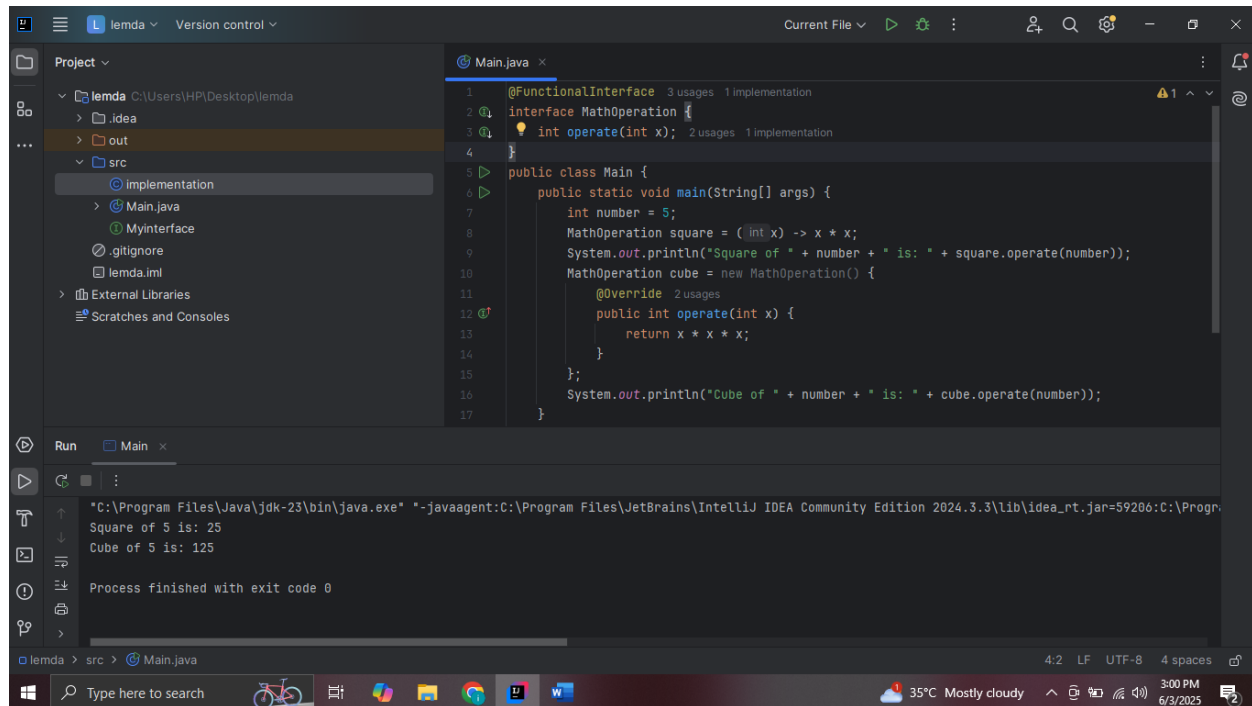


## Task 5 :

```
@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 :



```
1 @FunctionalInterface 3 usages 1 implementation
2 interface MathOperation {
3     int operate(int x); 2 usages 1 implementation
4 }
5
6 public class Main {
7     public static void main(String[] args) {
8         int number = 5;
9         MathOperation square = (int x) -> x * x;
10        System.out.println("Square of " + number + " is: " + square.operate(number));
11        MathOperation cube = new MathOperation() {
12            @Override 2 usages
13            public int operate(int x) {
14                return x * x * x;
15            }
16        };
17        System.out.println("Cube of " + number + " is: " + cube.operate(number));
18    }
19 }
```

Run Main x

```
"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=59206:C:\Progr..."
Square of 5 is: 25
Cube of 5 is: 125
Process finished with exit code 0
```

## Task 6 :

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

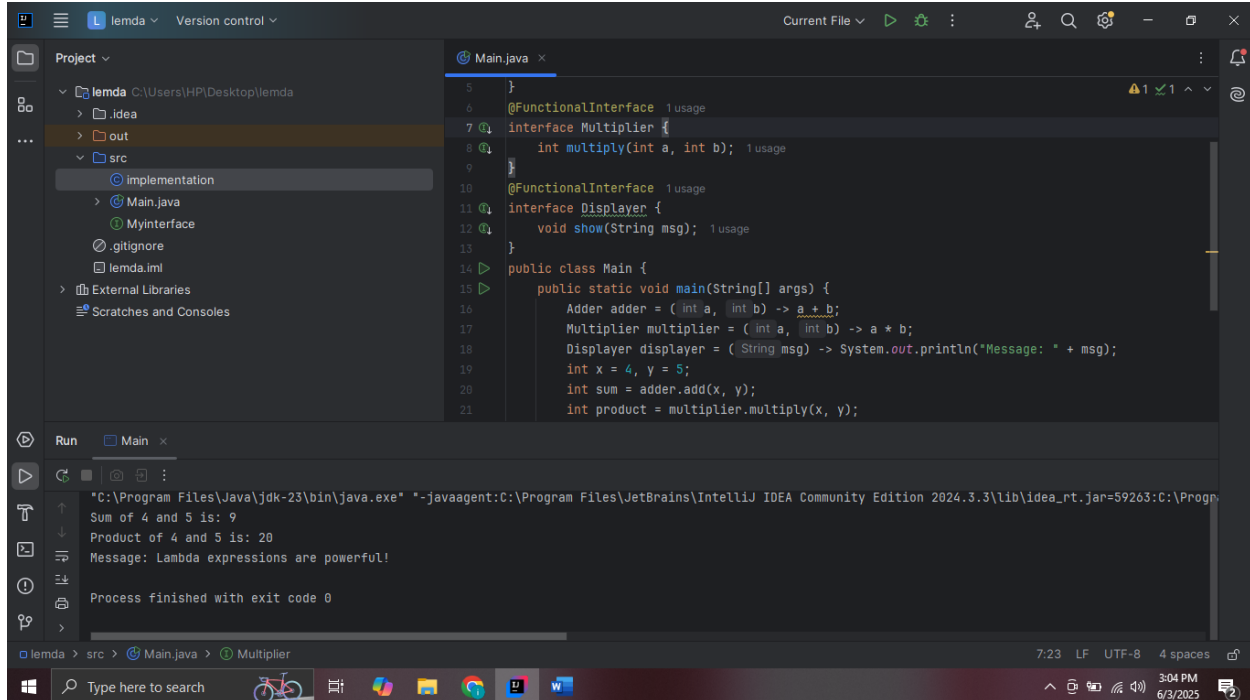
@FunctionalInterface
interface Multiplier {
    int multiply(int a, int b);
}

@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 :



The screenshot displays the IntelliJ IDEA IDE interface. The left sidebar shows the project structure for 'lemda', with 'src' expanded to show 'Main.java' and 'Myinterface'. The main editor window shows the code in 'Main.java', which includes two interfaces, 'Multiplier' and 'Displayer', and a 'Main' class. The 'Main' class contains a 'main' method that uses lambda expressions to create an 'Adder' and a 'Multiplier' object, and a 'Displayer' object to print messages. The bottom panel shows the output of the program, which includes the sum of 4 and 5, the product of 4 and 5, and a message about lambda expressions. The status bar at the bottom indicates the current file is 'Main.java' and the multiplier is 'Multiplier'.

```
5 }  
6 @FunctionalInterface 1 usage  
7 interface Multiplier {  
8     int multiply(int a, int b); 1 usage  
9 }  
10 @FunctionalInterface 1 usage  
11 interface Displayer {  
12     void show(String msg); 1 usage  
13 }  
14 public class Main {  
15     public static void main(String[] args) {  
16         Adder adder = (int a, int b) -> a + b;  
17         Multiplier multiplier = (int a, int b) -> a * b;  
18         Displayer displayer = (String msg) -> System.out.println("Message: " + msg);  
19         int x = 4, y = 5;  
20         int sum = adder.add(x, y);  
21         int product = multiplier.multiply(x, y);  
22     }  
23 }
```

Run Main x

```
"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:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.3\bin" -Didea.config.path=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.3\config 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 3:04 PM 6/3/2025