

## Week 2 – Day 2 Assignment (February 24th, 2026)

### Question 1 – Streams

IntelliJ IDEA 2020.3.3

Thu Feb 26 2:50 AM

homework > Stream.java

```
Employees average salary: 87600.0
Employees with salary > $80,000:
Sami
Landon
Bob

Employees map:
Landon -> Employee {Landon | 120000}
Bob -> Employee {Bob | 83000}
Leo -> Employee {Leo | 75000}
John -> Employee {John | 65000}
Sami -> Employee {Sami | 95000}

Employees name string: SamilandonLeoJohnBob

Process finished with exit code 0
```

IntelliJ IDEA 2020.3.3

Thu Feb 26 2:50 AM

homework > Employee.java

```
package questions.employee_streams;

import java.util.Objects;

public class Employee {
    private final String name;
    private final Integer salary;

    public Employee(String name, Integer salary) {
        this.name = Objects.requireNonNull(name, "name");
        this.salary = Objects.requireNonNull(salary, "salary");
    }

    public String getName() {
        return name;
    }

    public Integer getSalary() {
        return salary;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) {
            return true;
        }
        if (!(o instanceof Employee)) {
            return false;
        }
        Employee employee = (Employee) o;
        return Objects.equals(name, employee.name) && Objects.equals(salary, employee.salary);
    }

    @Override
    public int hashCode() {
        return Objects.hash(name, salary);
    }

    @Override
    public String toString() {
        return "Employee {" + name + " | " + salary + "}";
    }
}
```

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows the project structure under "homework".
- Code Editor:** Displays the content of `Stream.java`.
- Toolbars and Status Bar:** Includes standard IntelliJ IDEA toolbars and a status bar at the bottom indicating the file is 1122 lines long, uses LF line endings, is in UTF-8 encoding, and has 4 spaces.

```
1 package questions.employee_streams;
2
3 import java.sql.SQLOutput;
4 import java.util.ArrayList;
5 import java.util.Arrays;
6 import java.util.List;
7 import java.util.Map;
8 import java.util.function.Function;
9 import java.util.stream.Collectors;
10
11 public class Stream {
12     public static void main(String[] args) {
13         List<Employee> employees = new ArrayList<>();
14         employees.add(new Employee("Sami", salary: 95000));
15         employees.add(new Employee("Landon", salary: 128000));
16         employees.add(new Employee("Leo", salary: 75000));
17         employees.add(new Employee("John", salary: 65000));
18         employees.add(new Employee("Bob", salary: 85000));
19
20         /*
21          I) Calculate average salary of employees.
22         */
23         double averageSalary = employees.stream() Stream<Employee>
24             .mapToInt(Employee::getSalary) IntStream
25             .average() OptionalDouble
26             .orElse(0.0);
27         System.out.println("\nEmployees average salary: " + averageSalary);
28
29         /*
30          II) Get employees with salary > $80,000
31         */
32         System.out.println("\nEmployees with salary > $80,000:");
33         employees.stream() Stream<Employee>
34             .filter(e -> e.getSalary() > 80000)
35             .map(Employee::getName) Stream<String>
36             .forEach(System.out::println);
37
38         /*
39          III) Collect a map of employees storing name and salary (k, v).
40         */
41         Map<String, Employee> employeeMap = employees.stream()
42             .collect(Collectors.toMap(
43                 Employee::getName,
44                 Function.identity(),
45                 ( Employee e1, Employee e2 ) -> e1.getSalary() >= e2.getSalary() ? e1 : e2
46             ));
47     }
48 }
```

## Question 2 – Exceptions

IntelliJ IDEA screenshot showing the execution of a Java program.

The top window shows the output of the `Main.java` run:

```
/Library/Java/JavaVirtualMachines/jdk1.8.jdk/Contents/Home/bin/java ...
5
7 divided by 2 is not an integer.
Process finished with exit code 0
```

The bottom window shows the code editor for `NonIntResultException.java`:

```
package questions.my_exception;

public class NonIntResultException extends RuntimeException {
    private final int dividend; 2 usages
    private final int divisor; 2 usages

    public NonIntResultException(int dividend, int divisor) { 1 usage
        super(dividend + " divided by " + divisor + " is not an integer.");
        this.dividend = dividend;
        this.divisor = divisor;
    }

    public int getDividend() { no usages
        return dividend;
    }

    public int getDivisor() { no usages
        return divisor;
    }
}
```

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Bar:** homework > Main.java
- File Menu:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help
- Toolbar:** Current File, Version Control, etc.
- Project Explorer:** Shows the project structure with packages like homework, src, and my\_exception, and files like Main.java, NonIntResultException.java, .gitignore, and homework.html.
- Main.java Content:**

```
1 package questions.my_exception;
2
3 public class Main {
4     public static void main(String[] args) {
5         try {
6             System.out.println(divide(10, 2));
7             System.out.println(divide(7, 0));
8         } catch (NonIntResultException e) {
9             System.out.println(e.getMessage());
10        }
11    }
12
13    private static int divide(int a, int b) { 2 usages
14        if (b == 0) {
15            throw new ArithmeticException("Division by zero error.");
16        }
17
18        if (a % b != 0) {
19            throw new NonIntResultException(a, b);
20        }
21
22        return a / b;
23    }
24 }
```
- Status Bar:** homework > src > questions > my\_exception > Main > divide
- Bottom Right:** 20:10, LF, UTF-8, 8 spaces