

# Lab Report

By: Fikernew Birhanu (UGR/9932/13)

# Table of Contents

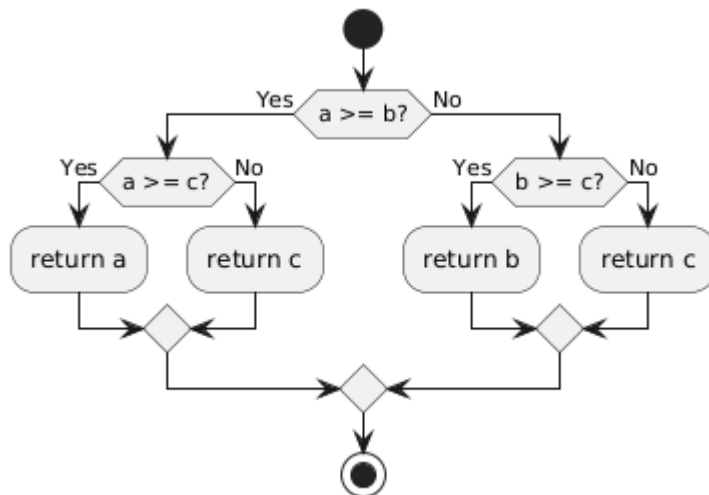
<b>Activity 1</b>	<b>2</b>
Code	2
CFG diagram	2
Complexity Calculation	2
Linearly independent paths	2
Path-based test cases	3
Test Results	3
<b>Activity 2</b>	<b>4</b>
Code	4
Test Cases	4
Coverage Report	4
<b>Activity 3</b>	<b>5</b>
Annotated Code	5
DU Path Graph	5
Test Cases	6
Test Results	6
<b>Activity 4</b>	<b>7</b>
Original code	7
Mutant versions	7
Test results	7
Mutation score	9
Mutmut (Automated Mutation Testing)	9
<b>Activity 5</b>	<b>11</b>
Java source code	11
JUnit test code	11
Test result output	11

# Activity 1

## Code

Code is in the file: `./src/activity_1/max_of_three_numbers.py`

## CFG diagram



## Complexity Calculation

To use the formula:  $C = E - N + 2P$ , we will need to find the number of edges, the number of nodes and connected components. For our function, we have:

- Edges: 11
- Nodes: 9
- Connected components: 1 (since we are evaluating a single function)

Therefore, the Cyclomatic Complexity will be 4.

## Linearly independent paths

The Cyclomatic Complexity of 4 indicates that there are 4 linearly independent paths through the code. Let's identify them:

- Path 1: Start → (a >= b is True) → (a >= c is True) → return a → End (Condition: a >= b and a >= c)
- Path 2: Start → (a >= b is True) → (a >= c is False) → return c → End (Condition: a >= b and a < c)

- Path 3: Start → (a >= b is False) → (b >= c is True) → return b → End (Condition: a < b and b >= c)
- Path 4: Start → (a >= b is False) → (b >= c is False) → return c → End (Condition: a < b and b < c)

## Path-based test cases

Based on the identified linearly independent paths, here are the corresponding test cases:

Path	Conditions	Input (a, b, c)	Expected Output	Rationale
1	a >= b AND a >= c	(5, 2, 3)	5	a is the maximum
2	a >= b AND a < c	(3, 2, 5)	5	c is the maximum, after initial a >= b is true (e.g., 3 vs 2)
3	a < b AND b >= c	(2, 5, 3)	5	b is the maximum, after initial a >= b is false (e.g., 2 vs 5)
4	a < b AND b < c	(2, 3, 5)	5	c is the maximum, after initial a >= b is false and b >= c is false

Test cases are in the file: `./src/activity_1/test_max_of_three_numbers.py`

## Test Results

```

===== tests coverage =====
coverage: platform darwin, python 3.13.3-final-0
Name                               Stmts  Miss Branch BrPart  Cover   Missing
-----
src/activity_1/__init__.py           0      0      0      0    100%
src/activity_1/max_of_three_numbers.py 8      0      6      0    100%
src/activity_1/test_max_of_three_numbers.py 24     0      0      0    100%
-----
TOTAL                               32      0      6      0    100%
Coverage HTML written to dir htmlcov
===== 6 passed in 0.07s =====

```

# Activity 2

## Code

Code is in the following file: `./src/activity_2/main.py`

## Test Cases

Test cases are in the following file: `./src/activity_2/test_main.py`

## Coverage Report

```
===== tests coverage =====
coverage: platform darwin, python 3.13.3-final-0
Name                               Stmts  Miss Branch BrPart  Cover  Missing
-----
src/__init__.py                     0      0      0      0    100%
src/activity_2/__init__.py          0      0      0      0    100%
src/activity_2/main.py              8      0      6      0    100%
src/activity_2/test_main.py         7      0      0      0    100%
-----
TOTAL                               15      0      6      0    100%
Coverage HTML written to dir htmlcov
===== 4 passed in 0.05s =====
```

# Activity 3

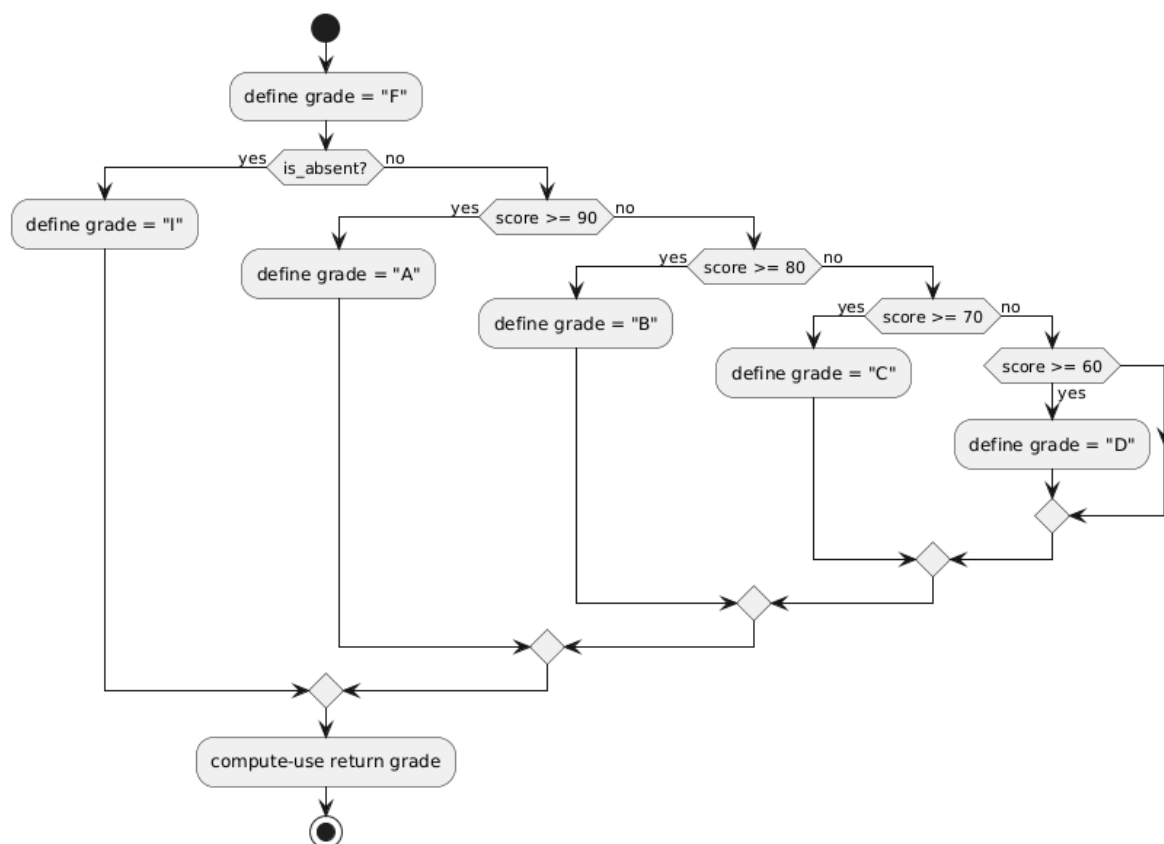
## Annotated Code

The annotated code is in the file: `./src/activity_3/annotated_code.py`

## DU Path Graph

Variables:

- grade
  - Definitions (d): "F", "I", "A", "B", "C", "D"
  - c-use: return grade
  - DU Paths: Each definition → return grade
- score
  - p-use:  $\geq 90$ ,  $\geq 80$ ,  $\geq 70$ ,  $\geq 60$
  - DU Paths: Parameter → each condition
- is\_absent
  - p-use: if is\_absent
  - DU Path: Parameter → condition



## Test Cases

Criterion	Test Input (score, is_absent)	Expected Output
All-defs	(50, False) → "F"	d1 → c-use
(95, True) → "I"	d2 → c-use	
(95, False) → "A"	d3 → c-use	
All DU Pairs	(80, False) → "B"	d → p-use (score ≥ 80)
(True, any) → "I"	d → p-use (is_absent)	
All DU Paths	(89, False) → "B"	Boundary (A/B)
(59, False) → "F"	Boundary (D/F)	

## Test Results

```

===== tests coverage =====
coverage: platform darwin, python 3.13.3-final-0
Name                               Stmts  Miss Branch BrPart  Cover   Missing
-----
src/activity_3/__init__.py           0      0      0      0    100%
src/activity_3/annotated_code.py     13      0     10      0    100%
src/activity_3/test_annotated_code.py 25      0      0      0    100%
-----
TOTAL                               38      0     10      0    100%
Coverage HTML written to dir htmlcov
===== 10 passed in 0.06s =====

```

# Activity 4

## Original code

Original code is in the following file: `./src/activity_4/original/rover.py`

## Mutant versions

Manual mutant versions are in the following folder: `./src/activity_4/mutants`

## Test results

Original test result:

```
> make test.activity.4.original
Make: Running activity tests...
Make: Running tests with coverage...
src/activity_4/original src/activity_4
python -m pytest --cov=src/activity_4/original --cov-report=term-missing --cov-branch --cov-report=html src/activity_4 --cov-fail-under=100 --ignore=**/_init__.py
===== test session starts =====
platform darwin -- Python 3.13.3, pytest-8.3.5, pluggy-1.6.0
rootdir: /Users/ffekirnew/code/school/sqa-lab3
configfile: pyproject.toml
plugins: cov-6.1.1
collected 8 items

src/activity_4/tests/test_rover.py ..... [100%]

===== tests coverage =====
coverage: platform darwin, python 3.13.3-final-0

Name                               Stmts  Miss Branch BrPart  Cover  Missing
-----
src/activity_4/original/_init__.py    0      0      0      0  100%
src/activity_4/original/rover.py     25      0     12      0  100%
-----
TOTAL                                25      0     12      0  100%
Coverage HTML written to dir htmlcov
Required test coverage of 100% reached. Total coverage: 100.00%
===== 8 passed in 0.05s =====
```

Mutant 1 test result:

```
===== tests coverage =====
coverage: platform darwin, python 3.13.3-final-0

Name                               Stmts  Miss  Cover
-----
src/activity_4/mutants/_init__.py    0      0  100%
src/activity_4/mutants/rover_mutant_1.py 27      3   89%
src/activity_4/mutants/rover_mutant_2.py 27     27    0%
src/activity_4/mutants/rover_mutant_3.py 27     27    0%
src/activity_4/mutants/rover_mutant_4.py 27     27    0%
src/activity_4/mutants/rover_mutant_5.py 27     27    0%
-----
TOTAL                                135    111   18%

===== short test summary info =====
FAILED src/activity_4/tests/test_rover.py::test_turn_left_from_north - AssertionError: assert <Direction.NORTH: 'N'> == <Direction.WEST: 'W'>
FAILED src/activity_4/tests/test_rover.py::test_turn_left_from_west - AssertionError: assert <Direction.WEST: 'W'> == <Direction.SOUTH: 'S'>
FAILED src/activity_4/tests/test_rover.py::test_turn_left_from_south - AssertionError: assert <Direction.WEST: 'W'> == <Direction.EAST: 'E'>
FAILED src/activity_4/tests/test_rover.py::test_turn_left_from_east - AssertionError: assert <Direction.WEST: 'W'> == <Direction.NORTH: 'N'>
===== 4 failed, 8 passed in 0.07s =====
make[1]: *** [test.file] Error 1
make: *** [test.activity.4.mutant] Error 2
```

Mutant 2 test result:



```

===== tests coverage =====
----- coverage: platform darwin, python 3.13.3-final-0 -----
Name                               Stmts  Miss  Cover
-----
src/activity_4/mutants/_init__.py    0      0  100%
src/activity_4/mutants/rover_mutant_1.py 27     27    0%
src/activity_4/mutants/rover_mutant_2.py 27      2   93%
src/activity_4/mutants/rover_mutant_3.py 27     27    0%
src/activity_4/mutants/rover_mutant_4.py 27     27    0%
src/activity_4/mutants/rover_mutant_5.py 27     27    0%
-----
TOTAL                               135    110   19%
===== short test summary info =====
FAILED src/activity_4/tests/test_rover.py::test_turn_left_from_west - AssertionError: assert <Direction.WEST: 'W'> == <Direction.SOUTH: 'S'>
FAILED src/activity_4/tests/test_rover.py::test_turn_left_from_south - AssertionError: assert <Direction.SOUTH: 'S'> == <Direction.EAST: 'E'>
FAILED src/activity_4/tests/test_rover.py::test_turn_left_from_east - AssertionError: assert <Direction.SOUTH: 'S'> == <Direction.NORTH: 'N'>
===== 3 failed, 9 passed in 0.08s =====
make[1]: *** [test.file] Error 1
make: *** [test.activity.4.mutant] Error 2

```

Mutant 3 test result:

```

===== tests coverage =====
----- coverage: platform darwin, python 3.13.3-final-0 -----
Name                               Stmts  Miss  Cover
-----
src/activity_4/mutants/_init__.py    0      0  100%
src/activity_4/mutants/rover_mutant_1.py 27     27    0%
src/activity_4/mutants/rover_mutant_2.py 27      2   93%
src/activity_4/mutants/rover_mutant_3.py 27     27    0%
src/activity_4/mutants/rover_mutant_4.py 27     27    0%
src/activity_4/mutants/rover_mutant_5.py 27     27    0%
-----
TOTAL                               135    110   19%
===== short test summary info =====
FAILED src/activity_4/tests/test_rover.py::test_turn_right_from_east - AssertionError: assert <Direction.EAST: 'E'> == <Direction.SOUTH: 'S'>
FAILED src/activity_4/tests/test_rover.py::test_turn_right_from_south - AssertionError: assert <Direction.SOUTH: 'S'> == <Direction.WEST: 'W'>
FAILED src/activity_4/tests/test_rover.py::test_turn_right_from_west - AssertionError: assert <Direction.SOUTH: 'S'> == <Direction.NORTH: 'N'>
===== 3 failed, 9 passed in 0.08s =====
make[1]: *** [test.file] Error 1
make: *** [test.activity.4.mutant] Error 2

```

Mutant 4 test result:

```

===== tests coverage =====
----- coverage: platform darwin, python 3.13.3-final-0 -----
Name                               Stmts  Miss  Cover
-----
src/activity_4/mutants/_init__.py    0      0  100%
src/activity_4/mutants/rover_mutant_1.py 27     27    0%
src/activity_4/mutants/rover_mutant_2.py 27     27    0%
src/activity_4/mutants/rover_mutant_3.py 27     27    0%
src/activity_4/mutants/rover_mutant_4.py 27      1   96%
src/activity_4/mutants/rover_mutant_5.py 27     27    0%
-----
TOTAL                               135    109   19%
===== short test summary info =====
FAILED src/activity_4/tests/test_rover.py::test_turn_right_from_west - AssertionError: assert <Direction.WEST: 'W'> == <Direction.NORTH: 'N'>
===== 1 failed, 11 passed in 0.07s =====
make[1]: *** [test.file] Error 1
make: *** [test.activity.4.mutant] Error 2

```

Mutant 5 test result:

```

===== tests coverage =====
----- coverage: platform darwin, python 3.13.3-final-0 -----
Name                               Stmts  Miss  Cover
-----
src/activity_4/mutants/_init_.py      0      0  100%
src/activity_4/mutants/rover_mutant_1.py 27     27   0%
src/activity_4/mutants/rover_mutant_2.py 27     27   0%
src/activity_4/mutants/rover_mutant_3.py 27     27   0%
src/activity_4/mutants/rover_mutant_4.py 27     27   0%
src/activity_4/mutants/rover_mutant_5.py 27      3  89%
-----
TOTAL                                135    111  18%
===== short test summary info =====
FAILED src/activity_4/tests/test_rover.py::test_turn_right_from_north - AssertionError: assert <Direction.NORTH: 'N'> == <Direction.EAS
T: 'E'>
FAILED src/activity_4/tests/test_rover.py::test_turn_right_from_east - AssertionError: assert <Direction.EAST: 'E'> == <Direction.SOUTH
: 'S'>
FAILED src/activity_4/tests/test_rover.py::test_turn_right_from_south - AssertionError: assert <Direction.EAST: 'E'> == <Direction.WEST
: 'W'>
FAILED src/activity_4/tests/test_rover.py::test_turn_right_from_west - AssertionError: assert <Direction.EAST: 'E'> == <Direction.NORTH
: 'N'>
===== 4 failed, 8 passed in 0.08s =====
make[1]: *** [test.file] Error 1
make: *** [test.activity_4.mutant] Error 2

```

## Mutation score

$$\text{Mutation Score} = \left( \frac{\text{Killed Mutants}}{\text{Total Mutants} - \text{Equivalent Mutants}} \right) \times 100$$

For my simple class and testing procedure, the mutation score is 100% as all 5 mutants were killed by the unit test.

## Mutmut (Automated Mutation Testing)

I used a python package called `mutmut` that will do automated mutation testing. After specifying to test the source code for activity for, we get the following result:

```

3 [tool.mutmut]
4 paths_to_mutate = [ "src/activity_4/original/" ]
5 tests_dir = [ "src/activity_4/original/" ]

> mutmut run
.: Generating mutants
  done in 59ms
.: Listing all tests
.: Running clean tests
  done
.: Running forced fail test
  done
Running mutation testing
.: 29/29 🚀 15 😊 14 🐞 0 🤔 0 😞 0 🦋 0
122.43 mutations/second

```

The code created by `mutmut` is found in the directory: `./mutants/src/activity_4`

The results show that `mutmut` has generated 29 mutations of our code and after running the unit tests, all of them have been cleared.

# Activity 5

## Java source code

The code is in the file: `./src/activity_5/main/java/com/example/Calculator.java`

## JUnit test code

The JUnit test code is in the file:

`./src/activity_5/main/java/com/example/CalculatorTest.java`

## Test result output

I run the whole project inside of a docker container. To do this I used the resources:

- Pom.xml (`./src/activity_5/pom.xml`)
- Dockerfile (`./src/activity_5/Dockerfile`)

```
# cat com.example.CalculatorTest.txt
-----
Test set: com.example.CalculatorTest
-----
Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.017 s - in com.example.CalculatorTest
```