Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik, Anja Rey

Exercise Sheet 1

for the lecture on

Advanced Programming and Algorithms

This exercise sheet contains the first problem to hand in. Please do so by uploading a PDF via ILIAS in order to get feedback. You can work in groups of up to three people. In order to pass the course, you'll need to pass at least 6 of the 12 hand-in problems.

Submission until **Monday, 23rd October, 12:30 pm**.
Discussion in the exercise classes on 30th October, 2nd and 3rd November, 2023.

Problem 1 to hand in: *Hand in*

The following pseudocode and Python code contain errors. For each subtask determine the error (i.e., explain what the error is) and write down how it can be repaired.

a) Given two positive integers a and b, the code should return the remainder c of the integer division $\lfloor \frac{a}{b} \rfloor$, i.e. $c = a - b \cdot \lfloor \frac{a}{b} \rfloor$.

`get_remainder(`a`, `b`)`:

```
1 while a > 0 do
2 |   a ← a − b
3 return a
```

b) Given an integer n, the following pseudocode should return a Boolean value `True` if and only if n is divisible by 3.

`is_even(n)`:

```
1 if n ≡ 0 mod 3 then
2 |   result ← True
3 result ← False
4 return result
```

c) Given two integers m and n, the following Python code should compute the power m^n.

```
1     def power(m, n):
2         power = 1
3         for index in range(n):
4             power = power * m
5             return power
```

d) Given a positive integer n, the following Python code should compute $\sum_{i=1}^{n} i^2$.

```
6     def sum_squares(n):
7         sum = 0
8         for index in range(n):
9             sum = sum + index**2
10        return sum
```

Problem 2 for discussion: *Efficiency and Refactoring*

Take a closer look at the code examples in Exercise 1.

a) What are their running times?

b) How can they be improved in terms of readability and brevity?

Problem 2 as a programming exercise: *Example Algorithm*

a) Implement the pseudocode example from the lecture (Lecture 1, slide 11) in Python.

b) What does it do? What happens in the background? Which computation problem does it solve?

c) How can the algorithm and the code be improved? What are useful quality criteria?