Programming Session I

Jan Tekülve
jan.tekuelve@ini.rub.de

Computer Science and Mathematics
Preparatory Course

30.09.2021
Brief Excursion on Number Systems

Number Systems

- **Natural Numbers**: \( \mathbb{N} = \{ 0, 1, 2, 3, 4, \ldots \} \)
- **Integer Numbers**: \( \mathbb{Z} = \ldots, -2, -1, 0, 1, 2, \ldots \)
- **Rational Numbers**: \( \mathbb{Q} \)
- **Real Numbers**: \( \mathbb{R} \)
Brief Excursion on Number Systems

Number Systems

- **Natural Numbers**: $\mathbb{N} = \{0, 1, 2, 3, 4, \ldots \}$
- **Integer Numbers**: $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots \}$
- **Rational Numbers**: $\mathbb{Q}$
- **Real Numbers**: $\mathbb{R}$
Brief Excursion on Number Systems

Number Systems

- **Natural Numbers**: \( \mathbb{N} = \{0, 1, 2, 3, 4, \ldots \} \)
- **Integer Numbers**: \( \mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots \} \)
- **Rational Numbers**: \( \mathbb{Q} = \frac{a}{b} \), where \( a, b \in \mathbb{Z} \) and \( b \neq 0 \)
- **Real Numbers**: \( \mathbb{R} \)
Real Numbers

- Between two rational numbers is an infinite amount of rational numbers

$\sqrt{2}$ is not a rational number.

$\sqrt{2} \approx 1.4142135\ldots$ is part of the real world.
Real Numbers

- Between two rational numbers is an infinite amount of rational numbers

- However: $\sqrt{2}$ is not a rational number
Real Numbers

- Between two rational numbers is an infinite amount of rational numbers
- However: $\sqrt{2}$ is not a rational number
- The irrational number $\sqrt{2} = 1.4142135...$ is part of the real world:
Definitions

Number Systems

- **Natural Numbers**: $\mathbb{N} = \{0, 1, 2, 3, 4, \ldots \}$
- **Integer Numbers**: $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots \}$
- **Rational Numbers**: $\mathbb{Q} = \frac{a}{b}$, where $a, b \in \mathbb{Z}$ and $b \neq 0$
- **Real Numbers**: $\mathbb{R} = \mathbb{Q} +$ irrational numbers
Definitions

Number Systems

- **Natural Numbers**: $\mathbb{N} = \{0, 1, 2, 3, 4, \ldots \}$
- **Integer Numbers**: $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots \}$
- **Rational Numbers**: $\mathbb{Q} = \frac{a}{b}$, where $a, b \in \mathbb{Z}$ and $b \neq 0$
- **Real Numbers**: $\mathbb{R} = \mathbb{Q} + \text{irrational numbers}$

Honorable Mention

- **Complex Numbers**: $\mathbb{C} = a + ib$, where $a, b \in \mathbb{R}$ and $i = \sqrt{-1}$
Number Systems

\[ \mathbb{R} \supseteq \mathbb{Q} \supseteq \mathbb{Z} \supseteq \mathbb{N} \]
Writing Files

- Opening a file
  
  #This creates the file if it does not exist
  fileObject = open("fileOutput.txt", "w")
  #Option 'w' will overwrite existing files
  #Use the option 'a' to append to a file instead

- Writing to the file
  
  #Add \n to end a line and \t to create a tab
  fileObject.write("Hello you!\n")

- Close the file after usage:
  
  fileObject.close()
If-Else

If and else are organized by indentation and colons

```python
x = 3.5
is_x_4 = False
if x == 4:  #if <condition> :
    is_x_4 = True  #indented block is called only
    print("x is 4")  #if <condition> applies
else:  #else is on the same level as if
    print("x is not 4")
#Regular program continues here
```
While Loops

- Print the numbers from 1 to 10

```python
goal = 5  # define two variables for the exit condition
test = 0

while test != goal:
    test = test + 1  # Increase test by 1
    print(test)  # prints 1,2,3,4,5 a number per loop
```
The List Datatype

- Lists allow to manage a collection of variables

```python
names = ['Alice','Bob','Carl','Dora']
numbers = [1,2,3,5,8]
```

- Accessing and modifying elements in a list

```python
print(names) # ['Alice','Bob','Carl','Dora']
single_name = names[2] # single_name = 'Carl'
first_element = numbers[0] # first_element = 1
last_name = names[len(names)-1] # last_name = 'Dora'

names[1] = 'Bert' # names ['Alice','Bert','Carl','Dora']
```
Operations on Lists

Example Operations

```python
numbers = [1,2,3,5,8]
names = ['Alice','Bob','Carl']
count = len(names) # count=3
names.append('Daisy') # ['Alice','Bob','Carl','Daisy']
numbers2 = [13,21,34]
numbers3 = numbers + numbers2 # [1,2,3,5,8,13,21,34]
subset = numbers3[2:5] # [3,5,8]
# characters from position 2 (included) to 5 (excluded)
```
Helpful Functions

➤ The random module

```python
import random  # import the module similar to import math
# assigns dice_roll a number between 1 and 6
dice_roll = random.randint(1,6)
# assigns coin_flip either a 0 or 1
coin_flip = random.randint(0,1)
```

➤ Deleting list elements

```python
names = ["alf","donald","charly brown","bud spencer"]
del names[1]  # deletes the second element
print(names)  # ["alf","charly brown","bud spencer"]
```
1. Write a Guessing Game, where the script chooses a random integer between 0 and 20 and the user has to guess it. With each guess the user gets told if his guess was higher or lower than the desired number.

- Start by assigning a random integer to a variable using `random.randint(0,20)`
- Create a while-loop in which the user is asked for a number
- Depending on the number input tell the user whether his guess was smaller, higher or equal to the desired value
- Think about how to end the while-loop
Tasks: Lists

2. Write a script that returns the biggest element in a list
   ▶ Create a list with arbitrary numbers of your choice
   ▶ Loop through the list with a for loop
   ▶ In each loop compare the current list element with your current estimate of the highest number

3. Write a script that looks for a specific element in the list and deletes it
   ▶ Loop through the list with a for-loop and store the elements position in a variable
   ▶ After the for loop remove the element at that position with the `del` command

4*. Write a script that takes a list and transfers its elements to a second list in sorted order.
   ▶ Look for the smallest element in the first list. Write it to the second list. Delete it in the first list. Repeat.
5*. Write a script that writes down the list from yesterday’s task 3 to a file:

- Start by opening the file
- First write “Coefficients:\n” to the file to create the first line
- Write your coefficients in the second line separated by commas
- Write “Values:” to the next line
- Run a loop through your list and in each loop write down x and the function value \(g(x)\) stored in the list

File Content Sketch:

<table>
<thead>
<tr>
<th>Coefficients:</th>
<th>(a_3), (a_2), (a_1), (a_0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0, (g(0))</td>
</tr>
<tr>
<td></td>
<td>1, (g(1))</td>
</tr>
<tr>
<td></td>
<td>\vdots</td>
</tr>
<tr>
<td></td>
<td>19, (g(19))</td>
</tr>
<tr>
<td></td>
<td>20, (g(20))</td>
</tr>
</tbody>
</table>