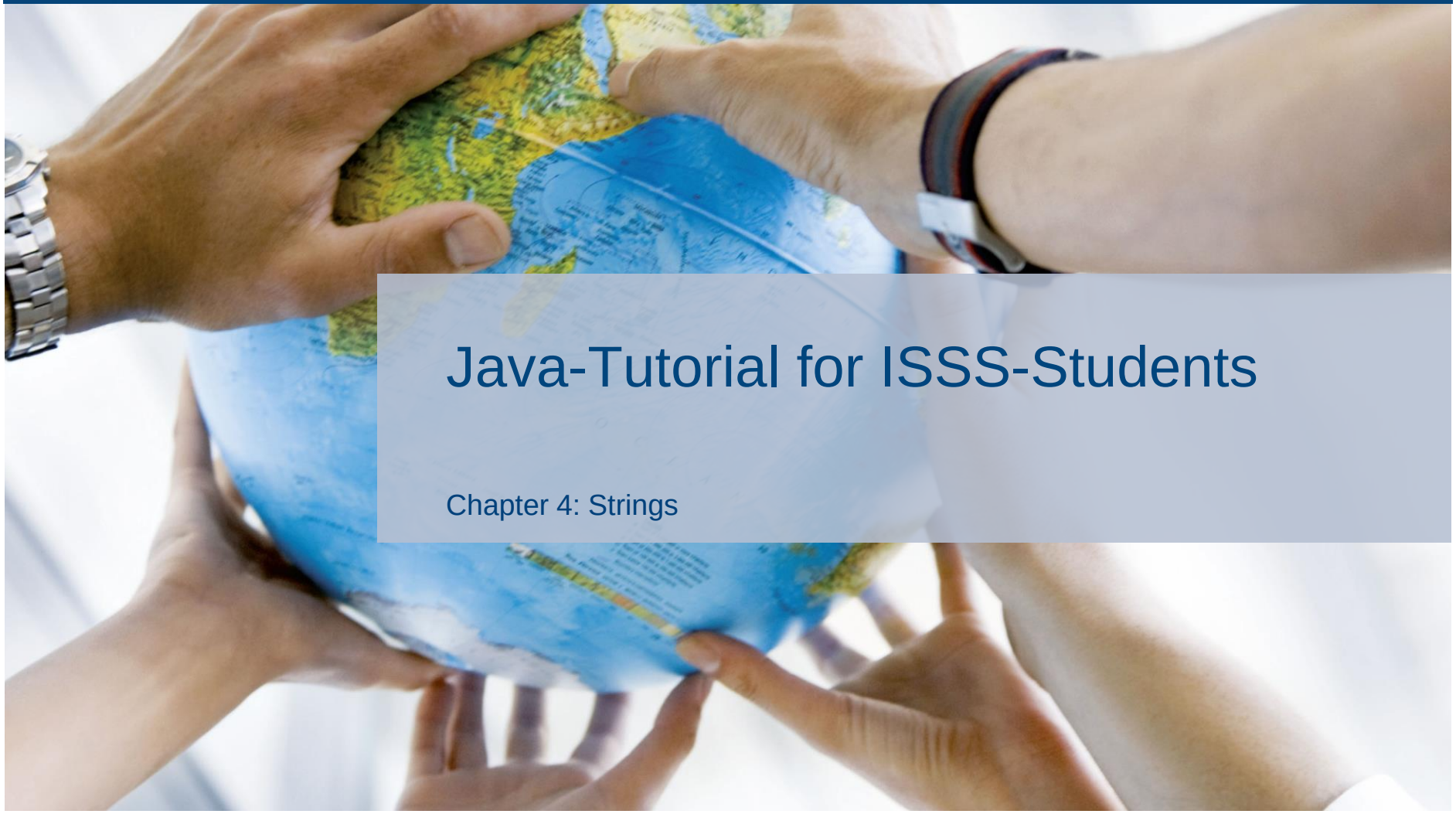
University of Bamberg



Java-Tutorial for ISSS-Students

Chapter 4: Strings

Chapter 4: Strings

1. Primitive vs. complex data types
2. Creating Strings
3. Using methods
4. Methods of the String class

Primitive vs. complex data types

- Primitive data types can be saved directly in variables:
  - *dataType variableName = value;*
    - ➢ Compare chapter 2
- Complex data types are implemented in *classes* and have to be instantiated in *objects* via a *constructor*:
  - *ObjectType variableName = new ObjectType();*
    - ➢ Compare chapter 5
- Strings are a special case in Java, as they are implemented in the class *String* but are often used similar to primitive data types

Creating Strings

- Via constructor:
  - *String s = new String(“Content”);*
- Via short form:
  - *String s = “Content”;*
- Via combining characters:
  - *char[] letters = {‘C’, ‘o’, ‘n’, ‘t’, ‘e’, ‘n’, ‘t’};*
  - *String s = new String(letters);*
- Via combining Strings:
  - *String s = “Con” + “tent”;*

Using methods

- While we work with primitive data types by using the given operators, complex data types contain their functionality in *methods*
- Methods are used by appending the method name to the variable name including the *parameters* needed for the method in brackets … :
  - *variableName.methodName(parameters);*
- … even if no parameters are needed at all:
  - *variableName.methodName();*

Methods of the String class

- *String s = "Further Content"*
  - Method *charAt()* returns the character at a given position:
    - *s.charAt(5) returns 'e'*
  - Method *length()* returns the length of the String:
    - *s.length() returns 15*
  - Method *substring()* returns a substring from a given index on … :
    - *s.substring(8) returns "Content"*
  - … or between two given indices:
    - *s.substring(5, 10) returns "er Co"*
  - Method *split()* returns a number of Strings resulting from splitting the original String at each position where a given String is found:
    - *s.split("e") returns "Furth", "r Cont" and "nt"*

Methods of the String class 2

- *String s = “Further Content”*
  - *equals()* returns true if the String equals another given String:
    - ➢ *s.equals(“Further Content”) returns true*
  - *startsWith()* returns true if the string starts with a given substring:
    - ➢ *s.startsWith(“Fur”) returns true*
  - *endsWith()* returns true if the string ends with a given substring:
    - ➢ *s.endsWith(“tent”) returns true*
  - *regionMatches()* returns true if the substring of a given String (specified by start index and length) is also a substring of the original String (starting from a specified index):
    - ➢ *s.regionMatches(9, “Montecarlo”, 1, 4) returns true*

Methods of the String class 3

- *String s = “Further Content”*
  - *indexOf()* returns the position of the first occurrence of a given character or substring in the whole String … :
    - *s.indexOf(‘t’) returns 3*
    - *s.indexOf(“Con”) returns 8*
  - … or from a given index on:
    - *s.indexOf(‘t’, 5) returns 11*
  - *contains()* returns true if the String contains a given substring:
    - *s.contains(“her”) returns true*
  - *replace()* replaces every occurrence of a given substring with another given substring:
    - *s.replace(“nt”, “lol”) returns “Further Cololelol”*