THE String CLASS

CSC 111

Outline

1. Class "String"
2. String Concatenation
   - 2.1 The '+' operator
3. String Declaration
4. String Initialization
   - 4.1 The new operator
5. Updating String variables

1. CLASS "String"

- A String is a sequence of zero or more characters.
- In Java, a String is enclosed between "double quotation".
- Students' names, Universities' names, Countries' names are stored in a String.
- Examples of strings include:
  - "Computer Science"
  - "King Saud University"
  - "KSA"
- Every character in a String has a specific position.
- The position of the first character in the String is zero.
- The length of a String is the number of included characters.

- A String that contains no characters is called a null string or an empty string. This is written as "".

The length of the empty (null) string is zero.

1. CLASS String

EXAMPLES

String	"How are you?"											
Character in the String	'H'	'o'	'w'	' '	'a'	'r'	'e'	' '	'y'	'o'	'u'	'?'
Position of the character	0	1	2	3	4	5	6	7	8	9	10	11

- Note that the space character ' ' has a position number.
- Also, special characters '?' have a position number.
- Maximum position = 11. Number of characters = 12.

String	"I am fine."									
Character in the String	'I'	' '	'a'	'm'	' '	'f'	'i'	'n'	'e'	'.'
Position of the character	0	1	2	3	4	5	6	7	8	9

- Maximum position = 9. Number of characters = 10.

Number of characters (length) = Maximum position + 1

2.1 String CONCATENATION

THE '+' OPERATOR

- String concatenation appends the second string to the first.
- The + operator is used to concatenate two strings.
- Examples:
  - "Programming with " + "Java I" = "Programming with Java I"
  - "My name is " + "Sara" = "My name is Sara"

- When a String is concatenated with a numeric value, the latter is converted into a String.
- Examples:
  - "Price is SR" + 28 = "Price is SR28"
  - "Pay rate is SR " + 30.5 = "Pay rate is SR 30.5"
  - "The sum is " + 12 + 24 = "The sum is 1224"
  - "The sum is " + (12 + 24) = The sum is 26

3. String DECLARATION

- The following statement declares a variable str of type String:

```
1  String str;
```

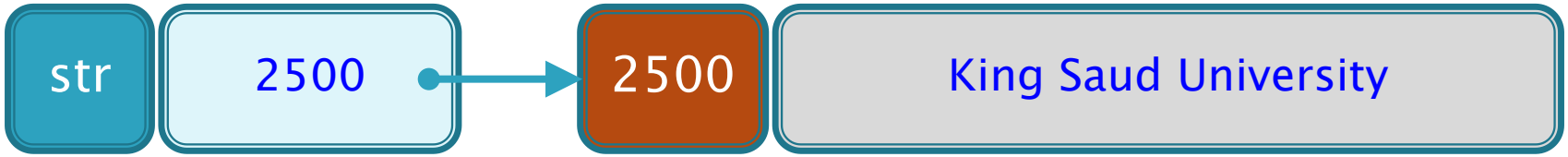
- In Java, String is NOT a primitive data type.
- A variable declared as a String class is known as a reference variable.
- A reference variable stores an address rather than a value.
- This is more illustrated in the next slide.

4. String INITIALIZATION

- The following statement initializes a variable str of type String:

```
1  String str = "King Saud University";
```

- After this statement, the memory layout is as follows:



- In the example above, 2500 is the address that stores "King Saud University".
- We say that str "points to" the memory location (address) that contains the string "King Saud University".
- The value of the memory location (address) - 2500 in this example - is specified by the operating system.
- Whenever we want to refer to "King Saud University", we just use the variable str.
- Since variables declared as String "refer to" a memory location, they are known as reference variables.

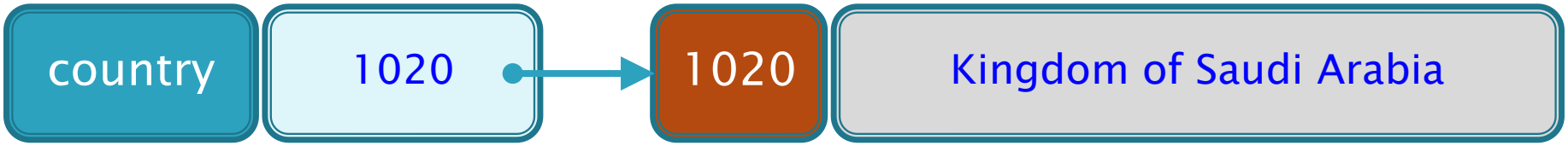
4.1 String INITIALIZATION

THE new OPERATOR

- The following statement also initializes a variable country of type String:

```
1  country = new String ("Kingdom of Saudi Arabia");
```

- After the above statement, the memory layout is as follows:



- The variable country "points to" or "refers to" the address 1020 that stores "Kingdom of Saudi Arabia"

- The above statement is identical to the following statement:

```
1  String country = "Kingdom of Saudi Arabia";
```

4.1 String INITIALIZATION

THE new OPERATOR

1 `country = new String ("Kingdom of Saudi Arabia");`

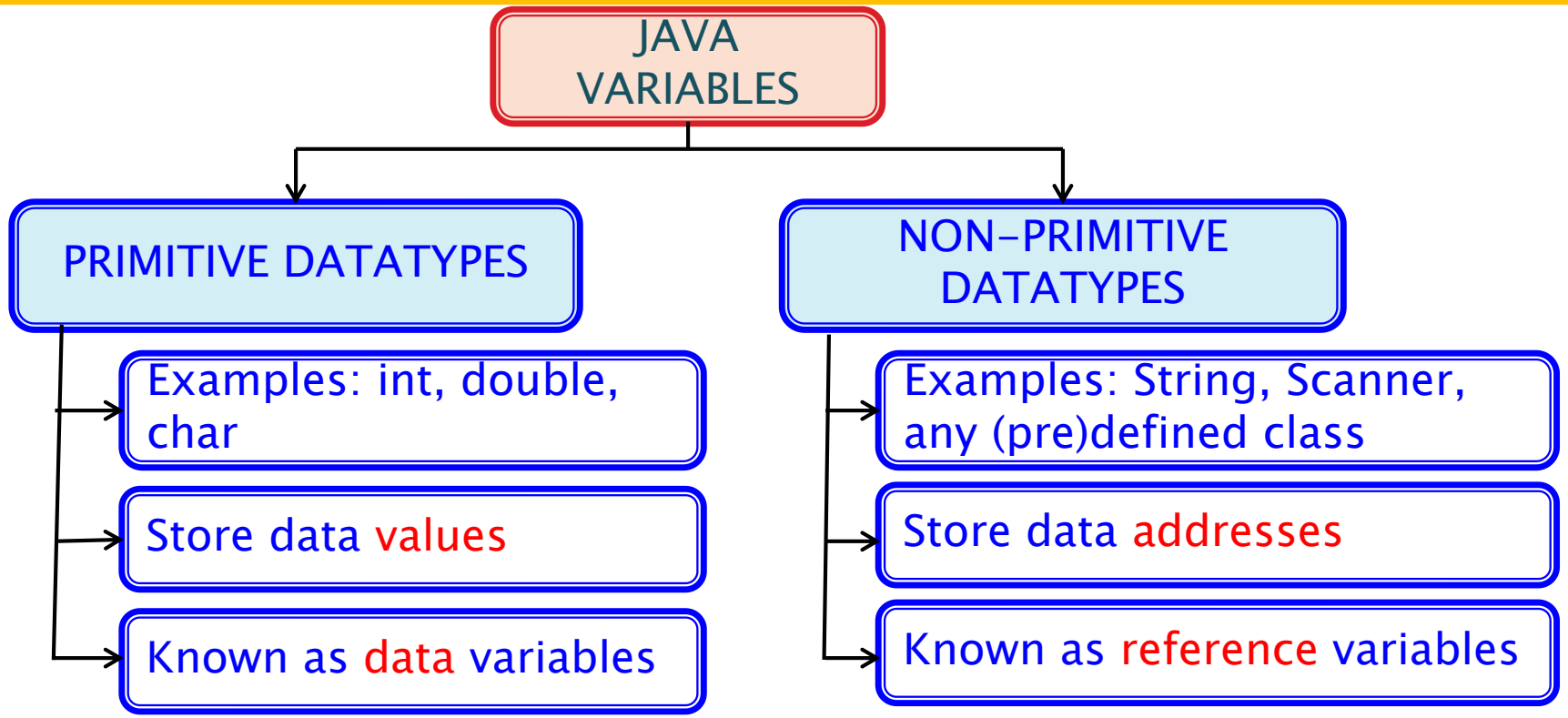
- In Java, the new operator causes the system to do the following:
  1) Allocate memory space (say 1020 in this example) of a specific type (String in this example)

  2) Store specific data (Kingdom of Saudi Arabia) in that memory space (1020)

  3) Return (store) the address of the memory space (1020) to the variable (country in this example)

5. JAVA VARIABLES

TO SUMMARIZE…



Self-Check Exercises

- Consider the following statements, and show the memory layout (trace) after each line:

```
1  String myFruit, fruit = "Orange";
2  String favorite = "Pineapple";
3  fruit = "Apple";
4  myFruit = favorite + fruit;
5  favorite = myFruit + fruit;
```