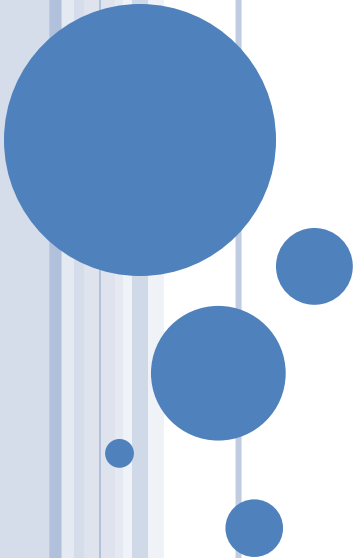




COMP 110 MORE TYPES

Instructor: Sasa Junuzovic

PREREQUISITES

- Types Math



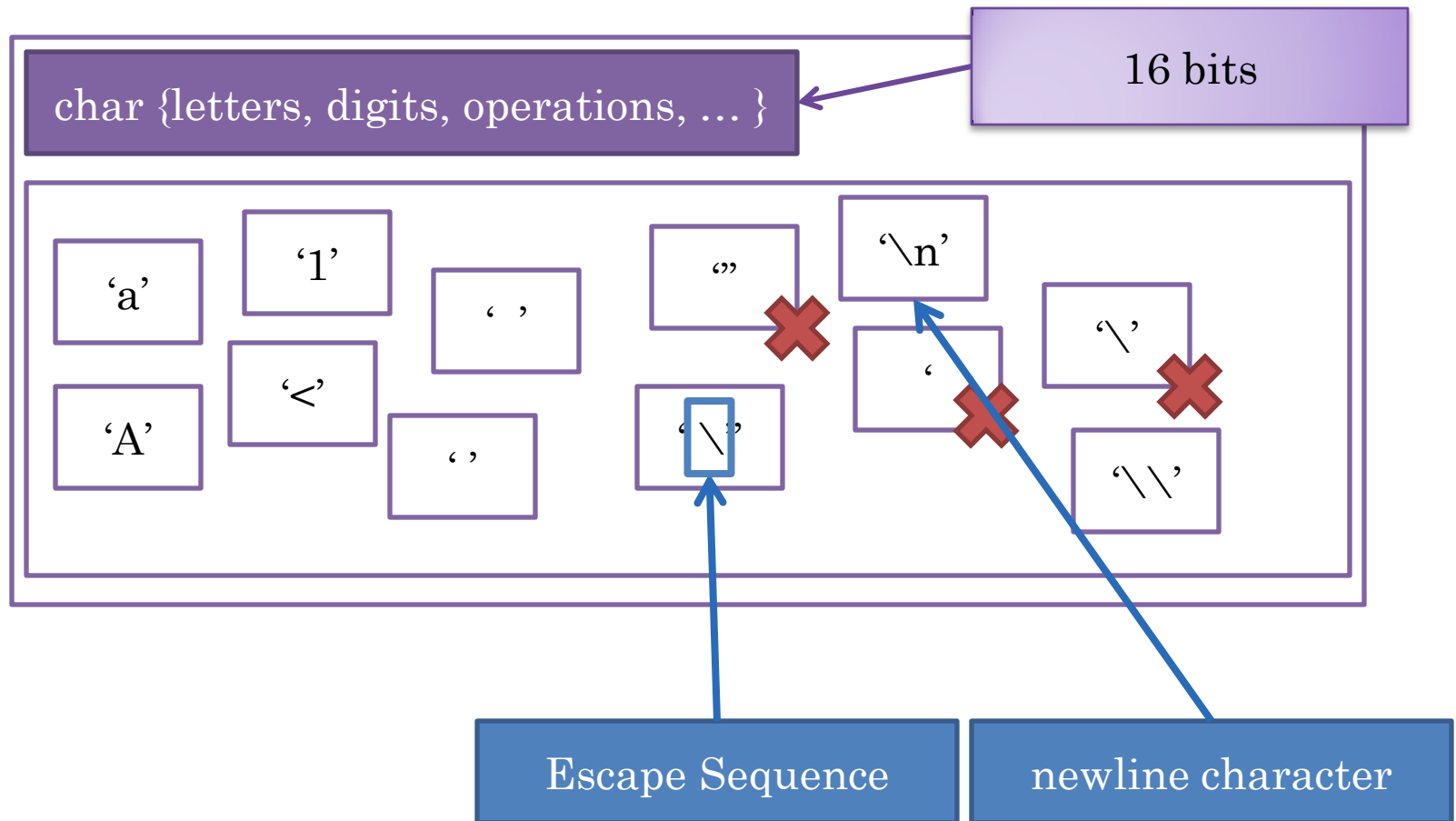
PRIMITIVE TYPES

- int, double, boolean, long, short, float, byte
- char

PRIMITIVE TYPES

- Constants (Literals & Named Constants)
- Operations with Invocation Syntax

CHAR CONSTANTS

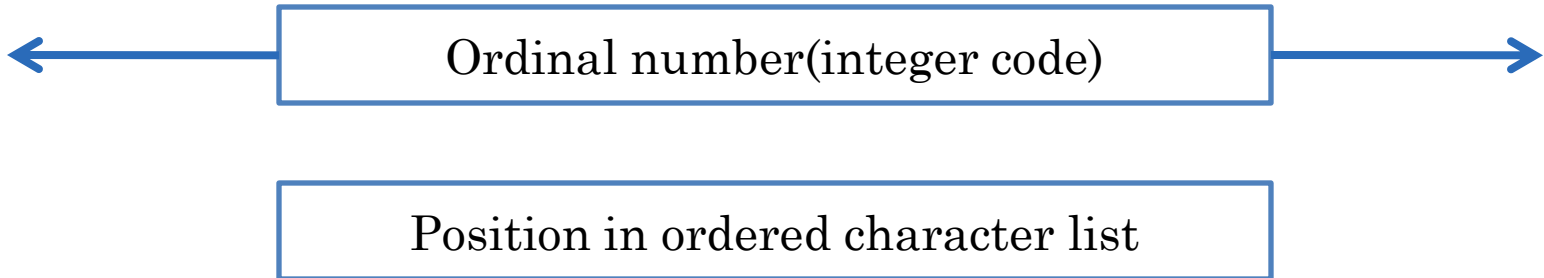


USEFUL ESCAPE SEQUENCES

Escape Sequence	Character Denoted
\'	'
\n	new line
\b	back space
\\	\
\t	tab
\"	"

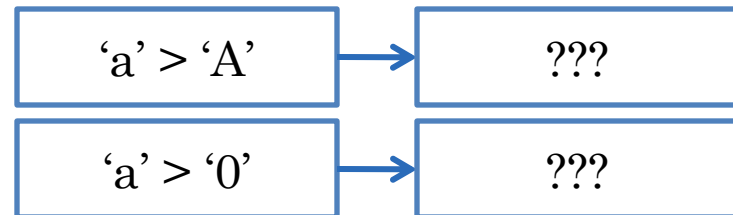
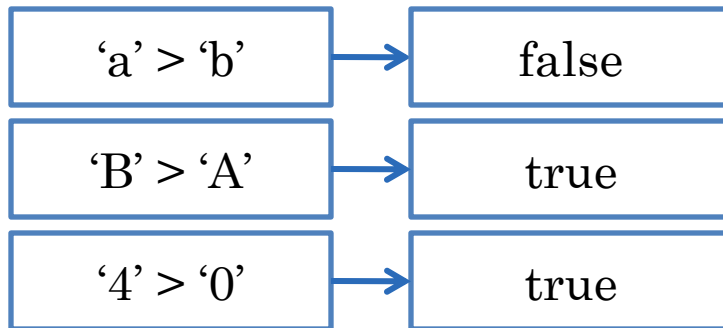
ORDERING CHARACTERS

“ ... ‘a’ ...



ORDERING CHARACTERS

“	...	‘a’ ‘b’ ‘c’	...
“	...	‘A’ ‘B’ ‘C’	...
“	...	‘0’ ‘1’ ‘2’	...



CONVERTING BETWEEN CHARACTERS AND THEIR ORDINAL NUMBERS

(int) 'a' → Ordinal number of 'a'

(char) 55 → Character whose ordinal number is 55

Implicit conversion to wider type

(int) " → 0

(char) 0 → "

(int) 'd' → ???

(char) 1 → ???

(char) -1 ❌

(int) 'c' - (int) 'a' → 2

'c' - 'a' → 2

(char) ('c' - 2) → 'a'

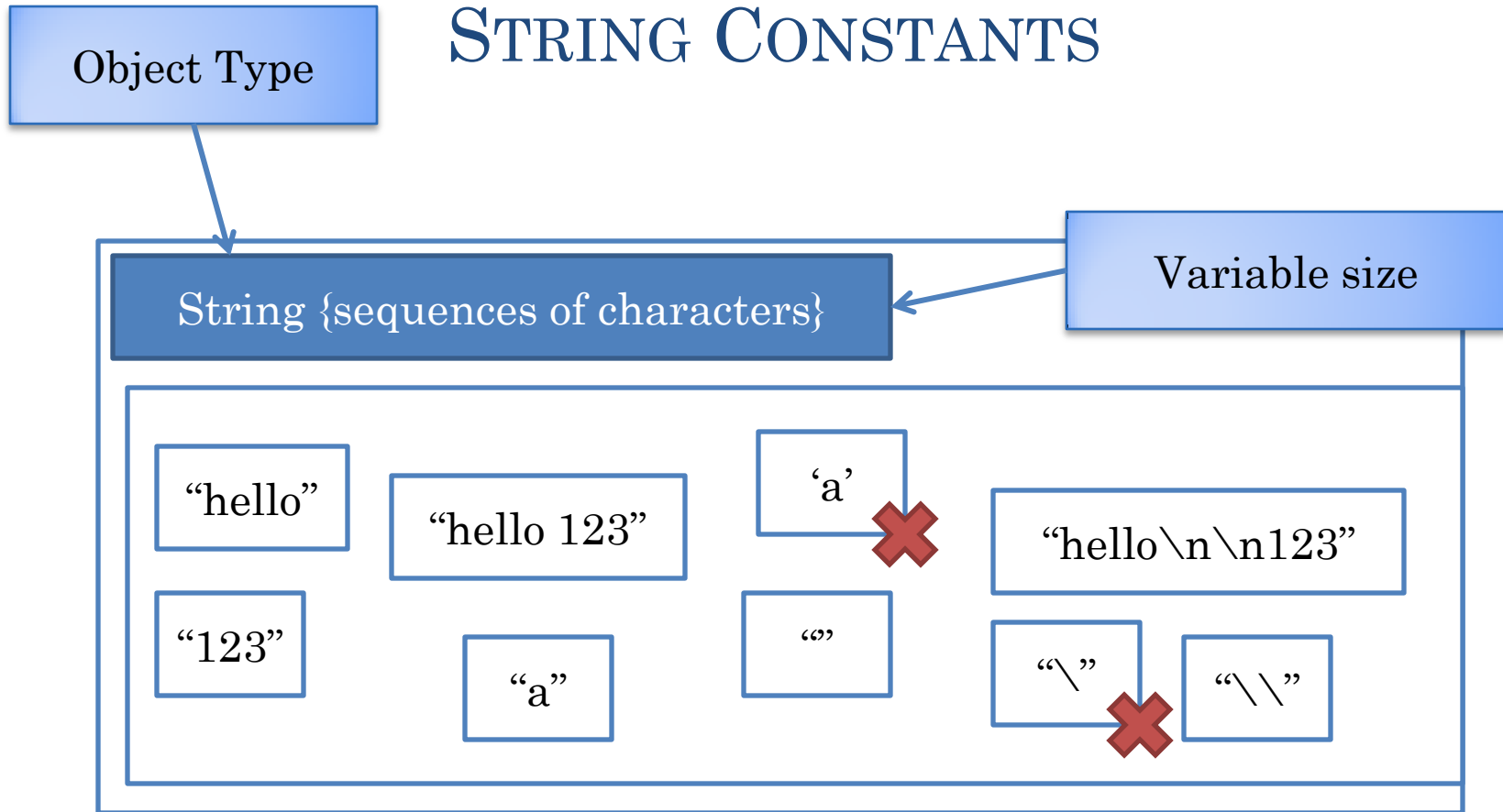
(char) ('A' + 2) → 'C'

(char) ('C' - 'A' + 'a') → 'c'

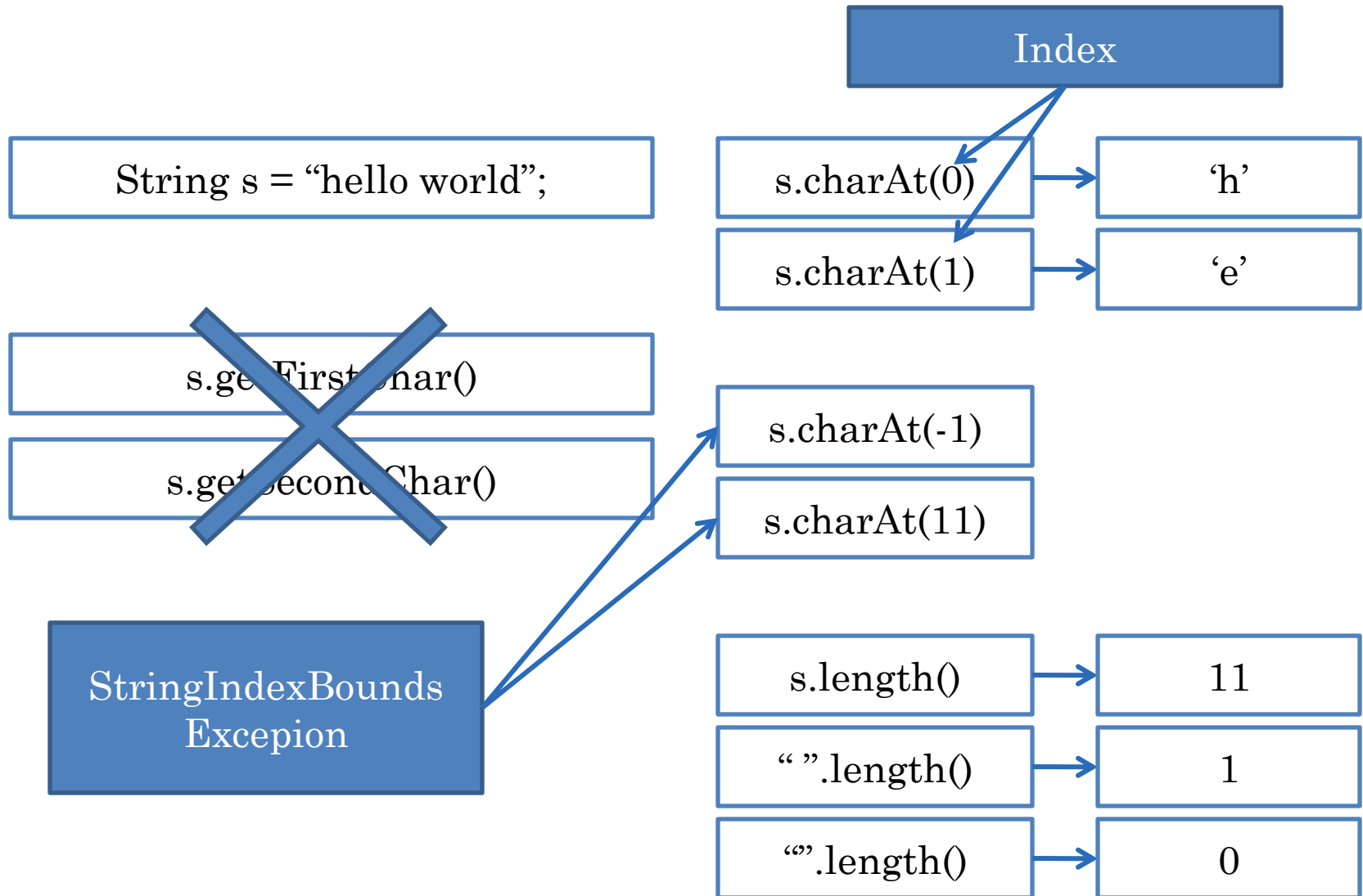
A USEFUL CHARACTER OPERATION

Character.isLetter(c)	→	true if c is a letter
Character.isLetter('c')	→	true
Character.isLetter('A')	→	true
Character.isLetter('1')	→	false
Character.isLetter(' ')	→	false

STRING CONSTANTS



ACCESSING STRING COMPONENTS



ACCESSING SUBSTRING

```
public String substring (int beginIndex, int endIndex)
```

```
s.substring(beginIndex, endIndex)
```

```
→ s.charAt(beginIndex) .. s.charAt(endIndex-1)
```

```
"hello world".substring(4,7)
```

```
"o w"
```

```
"hello world".substring(4,4)
```

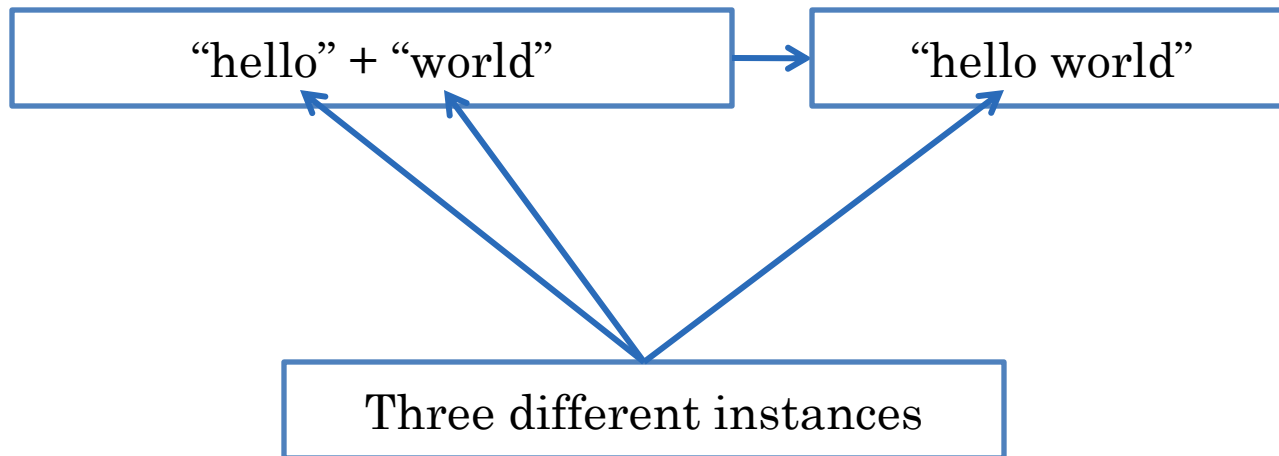
```
" "
```

```
"hello world".substring(7,4)
```

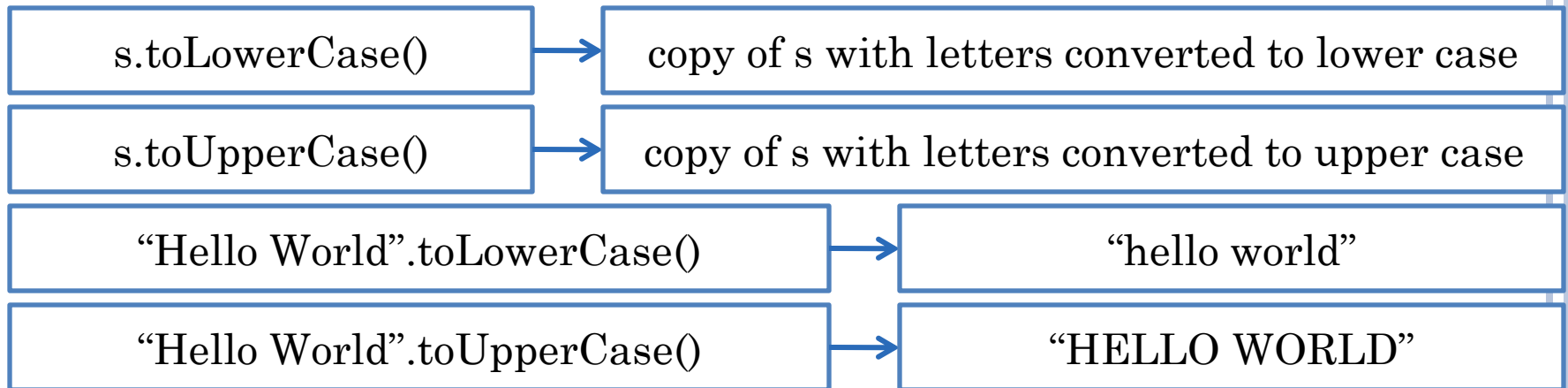
StringIndexBounds
Exception

CHANGING STRINGS?

Strings are read-only (immutable)



USEFUL STRING OPERATIONS



CLASSIFICATION

- Underweight
- Normal
- Overweight
- Obese
- Extremely Obese
- African Indian
- American Indian
- Asian
- Native Hawaiian
- White
- Some Other Race

Data type to store
a value capturing
the BMI
classification?

Data type to store
a value capturing
the Race?

INT CONSTANTS IN SEPARATE INTERFACES

```
public interface IntBMIClassification {  
    public static int UNDER_WEIGHT = 0;  
    public static int NORMAL = 1;  
    public static int OVER_WEIGHT = 2;  
    public static int OBESE = 3;  
    public static int EXTREMELY_OBESE = 4;  
}
```

```
public interface IntRace {  
    public static int AFRICAN_AMERICAN = 0;  
    public static int AMERICAN_INDIAN = 1;  
    public static int ASIAN = 2;  
    public static int NATIVE_HAWAIIAN = 4;  
    public static int WHITE = 4;  
    public static int SOME_OTHER_RACE = 5;  
}
```

IMPLEMENTATION WITH INT CONSTANTS

```
int race = IntRace.AFRICAN_AMERICAN;
public int getRace() {
    return race;
}
public void setRace (int newVal) {
    race = newVal;
}
public int getBMIClassification() {
    double bmi = getBMI();
    if (bmi <= MAX_UNDERWEIGHT_BMI)
        return BMIIntClassification.UNDER_WEIGHT;
    else if (bmi <= MAX_NORMAL_BMI)
        return BMIIntClassification.NORMAL;
    else if (bmi <= MAX_OVERWEIGHT_BMI)
        return BMIIntClassification.OVER_WEIGHT;
    else if (bmi <= MAX_OBESE_BMI)
        return BMIIntClassification.OBESE;
    else
        return BMIIntClassification.EXTREMELY_OBESE;
}
```

PROGRAMMER ERRORS POSSIBLE

```
public interface BMIIntClassification {  
    public static int UNDER_WEIGHT = 0;  
    public static int NORMAL = 1;  
    public static int OVER_WEIGHT = 2;  
    public static int OBESE = 3;  
    public static int EXTREMELY_OBESE = 3;  
}
```

Programming language does not
know the relationship among
constants!

UNINFORMATIVE OUTPUT AND USER ERRORS

Common	
Height:	1.77
Race:*	5
Weight:	25.0
BMI:	23.93948099205209
BMI Classification:	1

User can make mistake
in assigning a value

STRING CONSTANTS

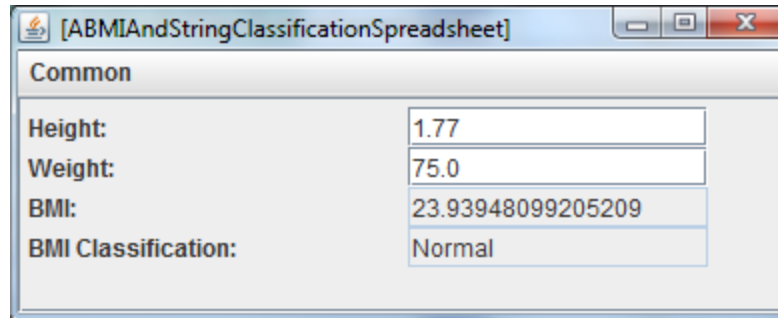
```
public interface StringBMIClassification {  
    public static String UNDER_WEIGHT = "Under Weight";  
    public static String NORMAL = "Normal";  
    public static String OVER_WEIGHT = "Over Weight";  
    public static String OBESE = "Obese";  
    public static String EXTREMELY_OBESE = "Extremely Obese";  
}
```

```
public interface StringRace {  
    public static String AFRICAN_AMERICAN = "African American";  
    public static String AMERICAN_INDIAN = "American Indian";  
    public static String ASIAN = "Asian";  
    public static String NATIVE_HAWAIIAN = "Native Hawaiian";  
    public static String WHITE = "White";  
    public static String SOME_OTHER_RACE = "Some Other Race";  
}
```

IMPLEMENTATION WITH STRING CONSTANTS

```
String race = StringRace.AFRICAN_AMERICAN;
public String getRace() {
    return race;
}
public void setRace (String newVal) {
    race = newVal;
}
public String getBMIClassification() {
    double bmi = getBMI();
    if (bmi <= MAX_UNDERWEIGHT_BMI)
        return StringBMIClassification.UNDER_WEIGHT;
    else if (bmi <= MAX_NORMAL_BMI)
        return StringBMIClassification.NORMAL;
    else if (bmi <= MAX_OVERWEIGHT_BMI)
        return StringBMIClassification.OVER_WEIGHT;
    else if (bmi <= MAX_OBESE_BMI)
        return StringBMIClassification.OBESE;
    else
        return StringBMIClassification.EXTREMELY_OBESE;
}
```

ERRORS POSSIBLE



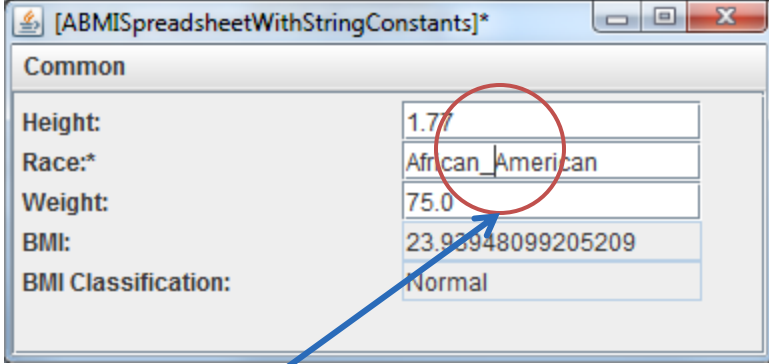
A screenshot of a spreadsheet application window titled "[ABMIAndStringClassificationSpreadsheet]". The window displays a "Common" tab with four rows of data. The first two rows are input fields for "Height" and "Weight". The third row shows the calculated "BMI" value. The fourth row shows the "BMI Classification" result.

Common	
Height:	1.77
Weight:	75.0
BMI:	23.93948099205209
BMI Classification:	Normal

String type is space inefficient

User or programmer can make mistake in assigning a value

STRING ISSUES



Common	
Height:	1.77
Race:*	African_American
Weight:	75.0
BMI:	23.93948099205209
BMI Classification:	Normal

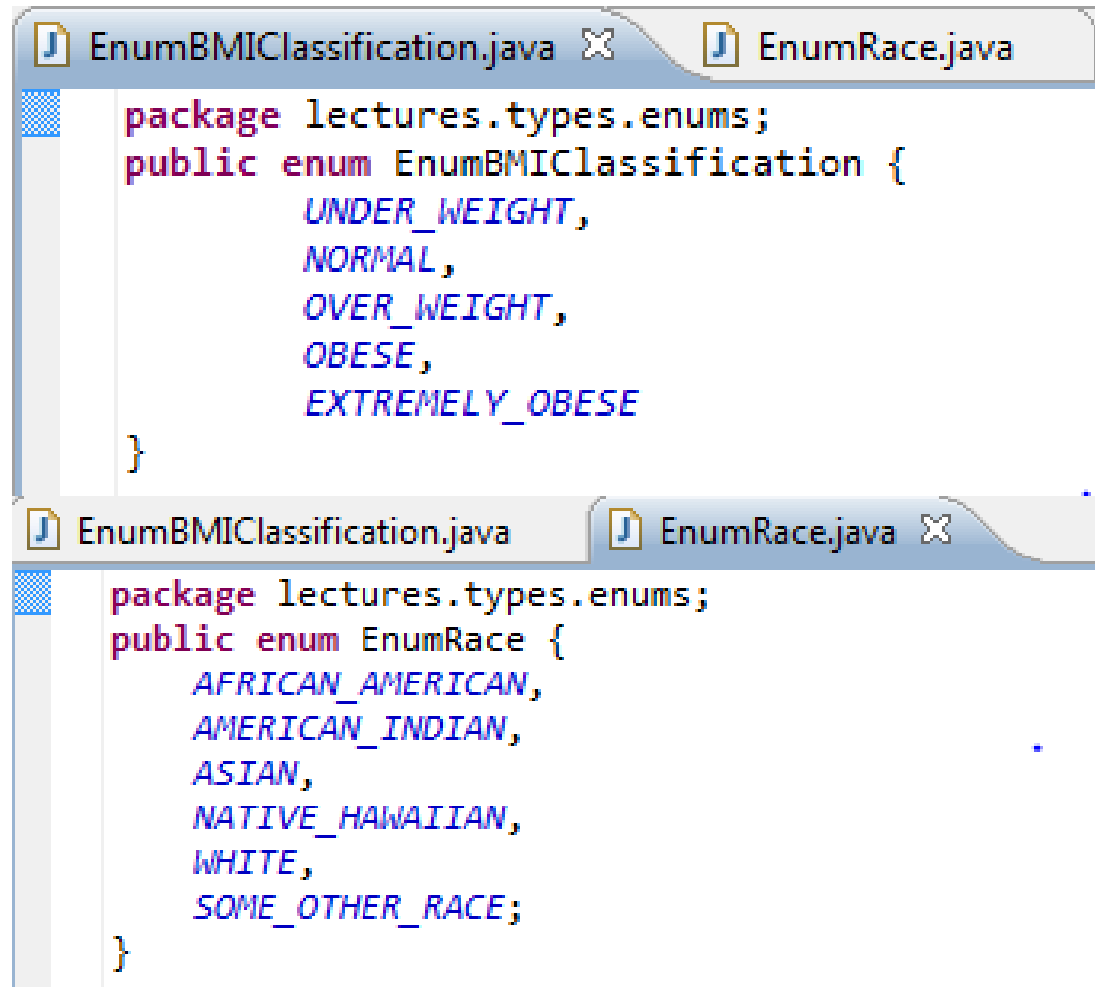
User or programmer can make mistake in assigning a value

String type is space inefficient

ENUM

- Like a string set, a set of named choices
- Like an int set, each choice is a number

DECLARING AN ENUM



```
EnumBMIClassification.java
package lectures.types.enums;
public enum EnumBMIClassification {
    UNDER_WEIGHT,
    NORMAL,
    OVER_WEIGHT,
    OBESE,
    EXTREMELY_OBESE
}

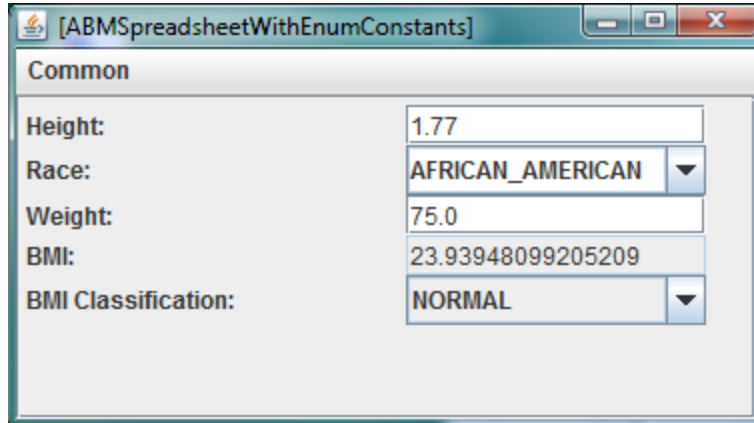
EnumRace.java
package lectures.types.enums;
public enum EnumRace {
    AFRICAN_AMERICAN,
    AMERICAN_INDIAN,
    ASIAN,
    NATIVE_HAWAIIAN,
    WHITE,
    SOME_OTHER_RACE;
}
```

Like class or an interface, an enum is declared in its own file

IMPLEMENTATION WITH STRING CONSTANTS

```
EnumRace race = EnumRace.AFRICAN_AMERICAN;
public EnumRace getRace() {
    return race;
}
public void setRace (EnumRace newVal) {
    race = newVal;
}
public EnumBMIClassification getBMIClassification() {
    double bmi = getBMI();
    if (bmi <= MAX_UNDERWEIGHT_BMI)
        return EnumBMIClassification.UNDER_WEIGHT;
    else if (bmi <= MAX_NORMAL_BMI)
        return EnumBMIClassification.NORMAL;
    else if (bmi <= MAX_OVERWEIGHT_BMI)
        return EnumBMIClassification.OVER_WEIGHT;
    else if (bmi <= MAX_OBESE_BMI)
        return EnumBMIClassification.OBESE;
    else
        return EnumBMIClassification.EXTREMELY_OBESE;
}
```

DISPLAYING AND EDITING ENUM



[ABMSpreadsheetWithEnumConstants]

Common

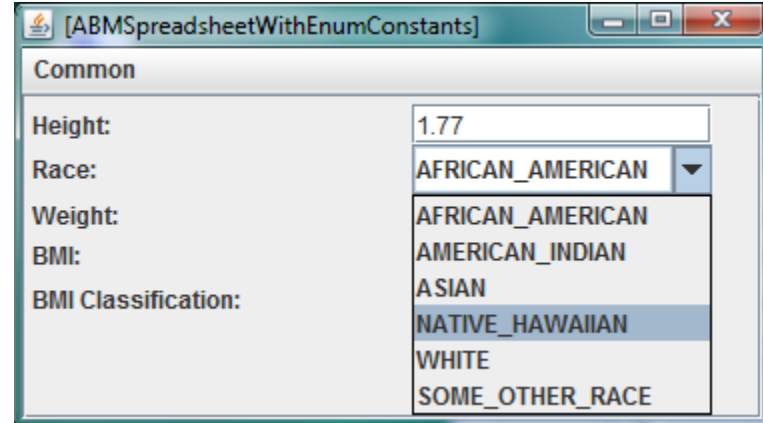
Height: 1.77

Race: AFRICAN_AMERICAN ▼

Weight: 75.0

BMI: 23.93948099205209

BMI Classification: NORMAL ▼



[ABMSpreadsheetWithEnumConstants]

Common

Height: 1.77

Race: AFRICAN_AMERICAN ▼

Weight: 75.0

BMI: 23.93948099205209

BMI Classification: AFRICAN_AMERICAN
AMERICAN_INDIAN
ASIAN
NATIVE_HAWAIIAN
WHITE
SOME_OTHER_RACE