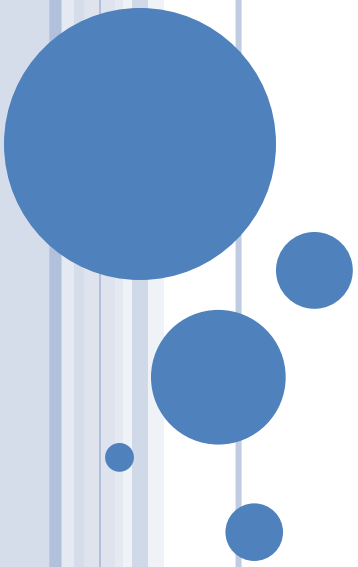


Comp 110
More Types

Instructor: Sasa Junuzovic

Prerequisites

- Types Math

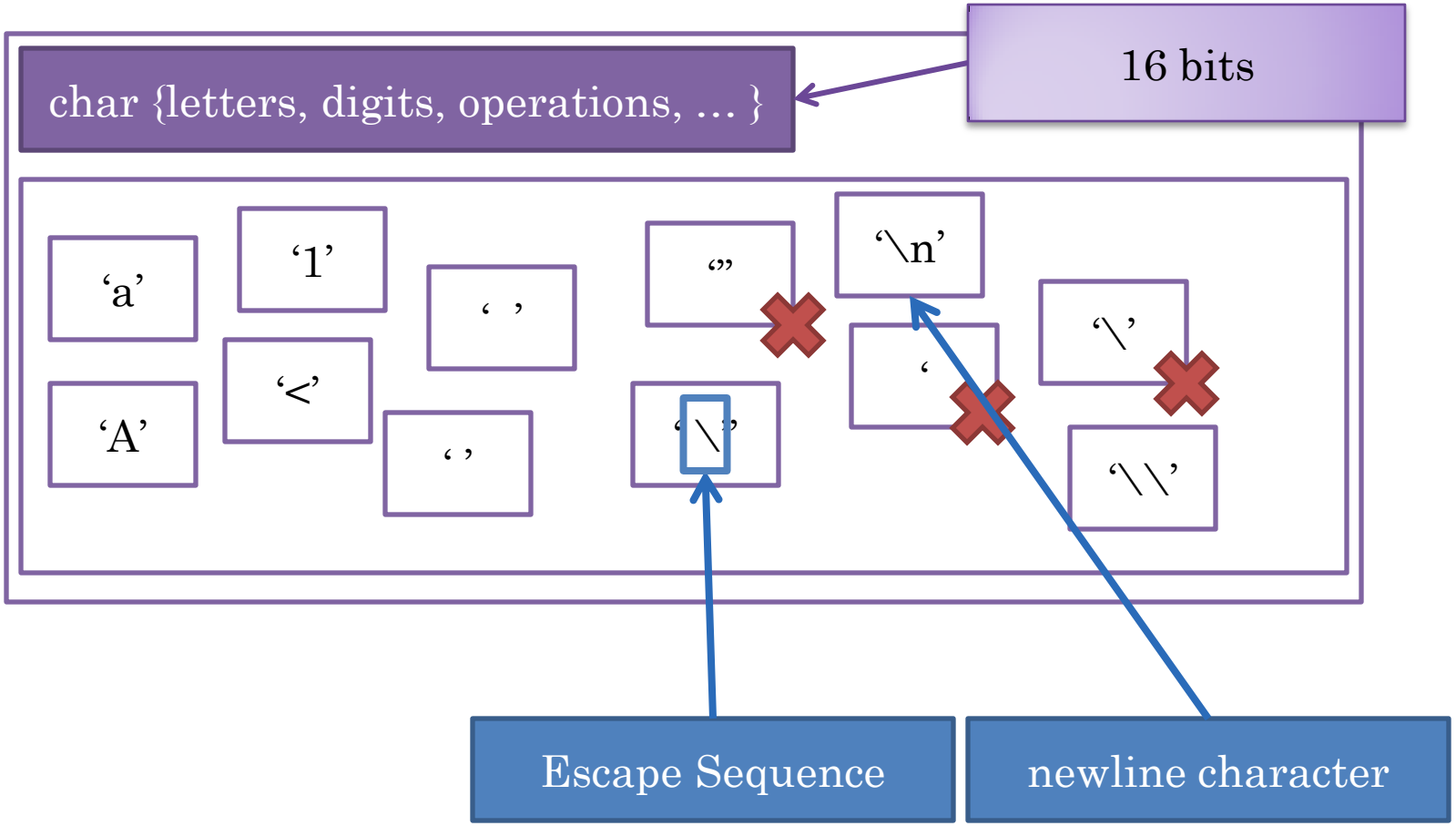
Primitive Types

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

Primitive Types

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

CHAR CONSTANTS

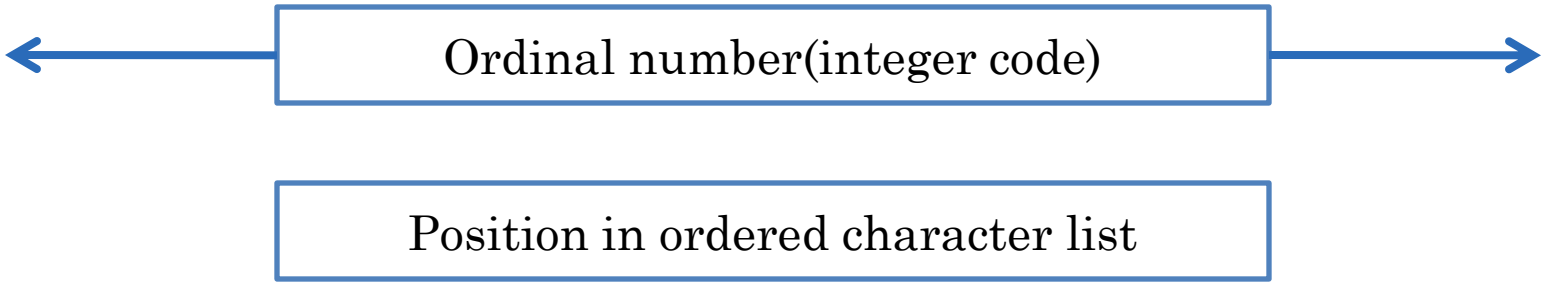


Useful Escape Sequences

Escape Sequence	Character Denoted
\’	’
\n	new line
\b	back space
\\	\
\t	tab
\”	”

Ordering Characters

‘ … ‘a’ …



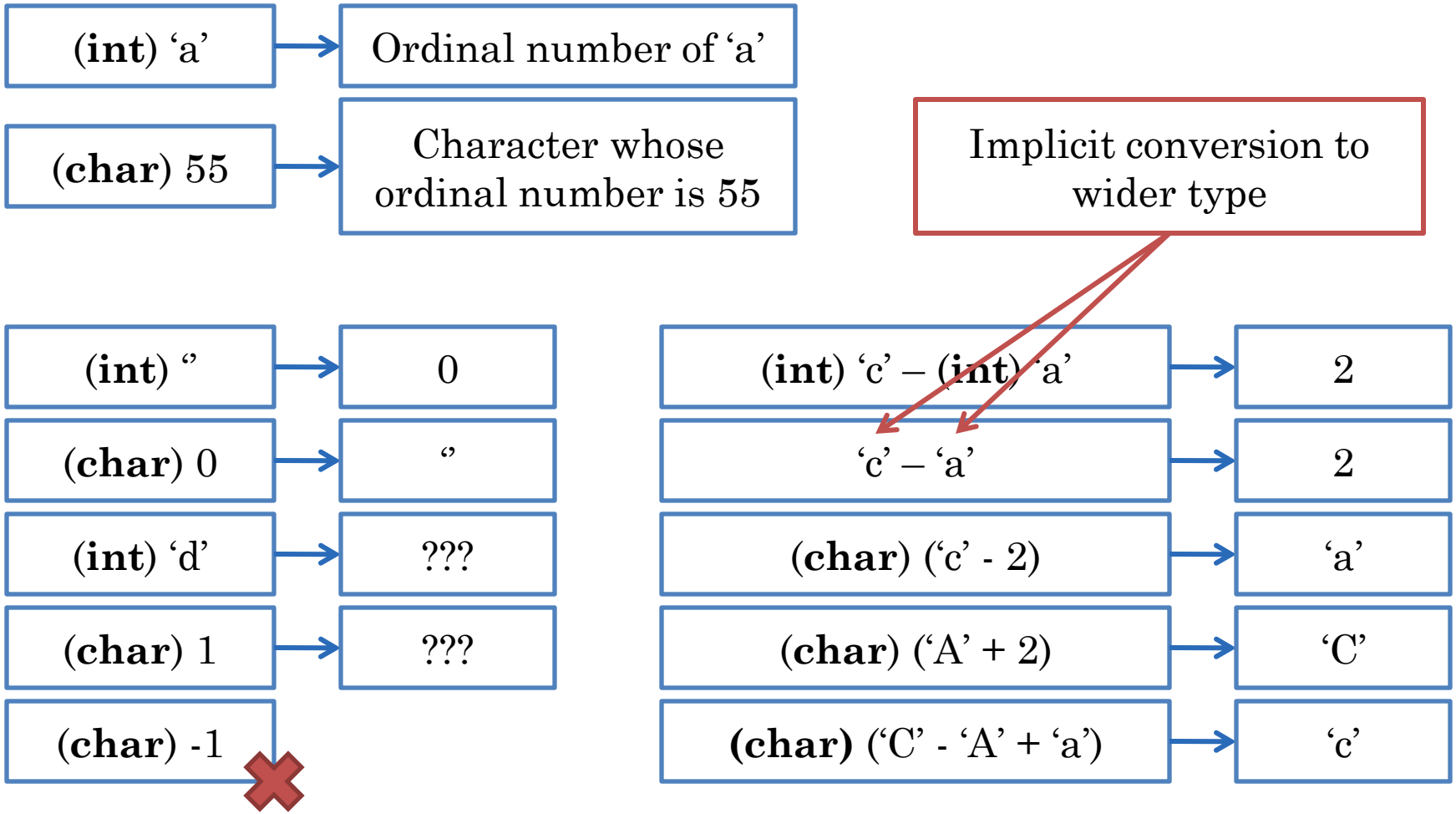
Ordering Characters

‘’	…	‘a’ ‘b’ ‘c’	…
‘’	…	‘A’ ‘B’ ‘C’	…
‘’	…	‘0’ ‘1’ ‘2’	…

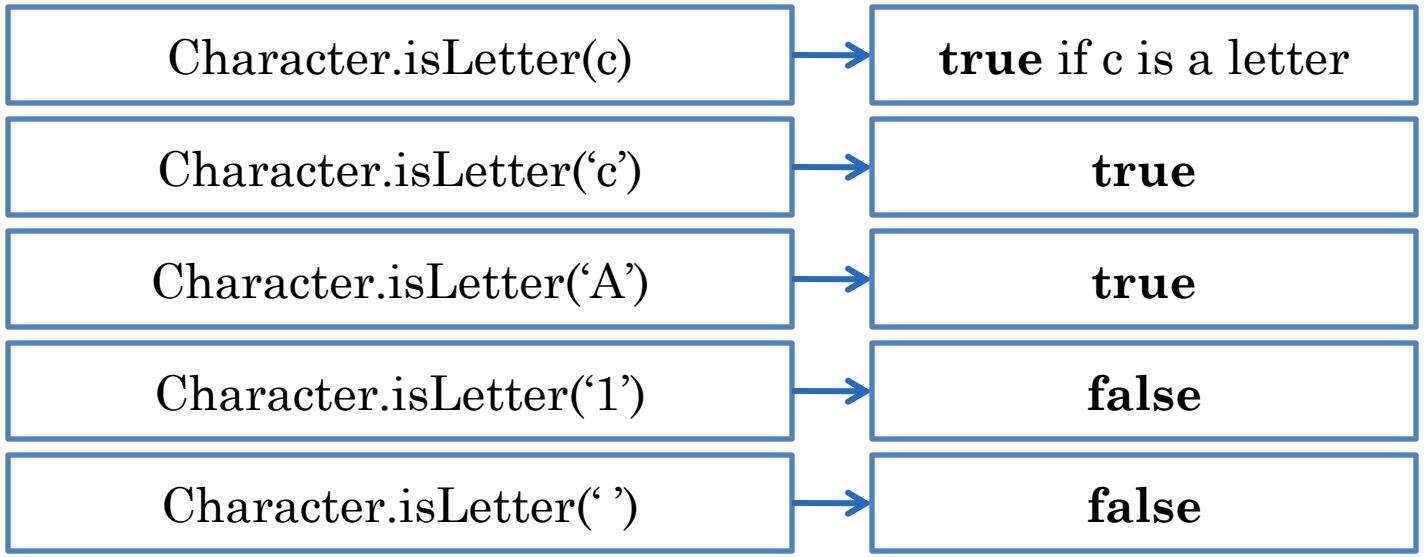
‘a’ > ‘b’	false
‘B’ > ‘A’	true
‘4’ > ‘0’	true

‘a’ > ‘A’	???
‘a’ > ‘0’	???

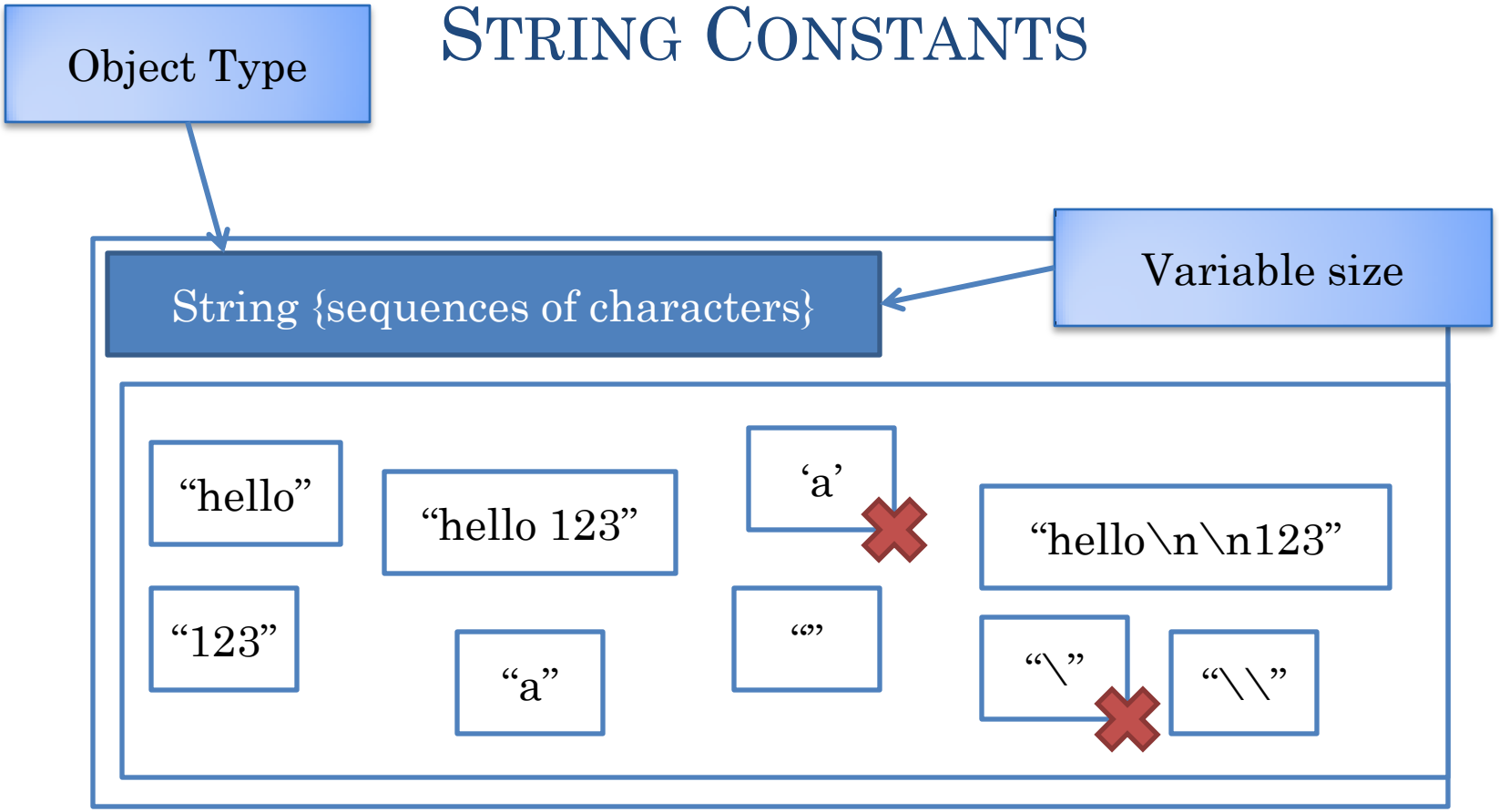
Converting Between Characters and Their Ordinal Numbers



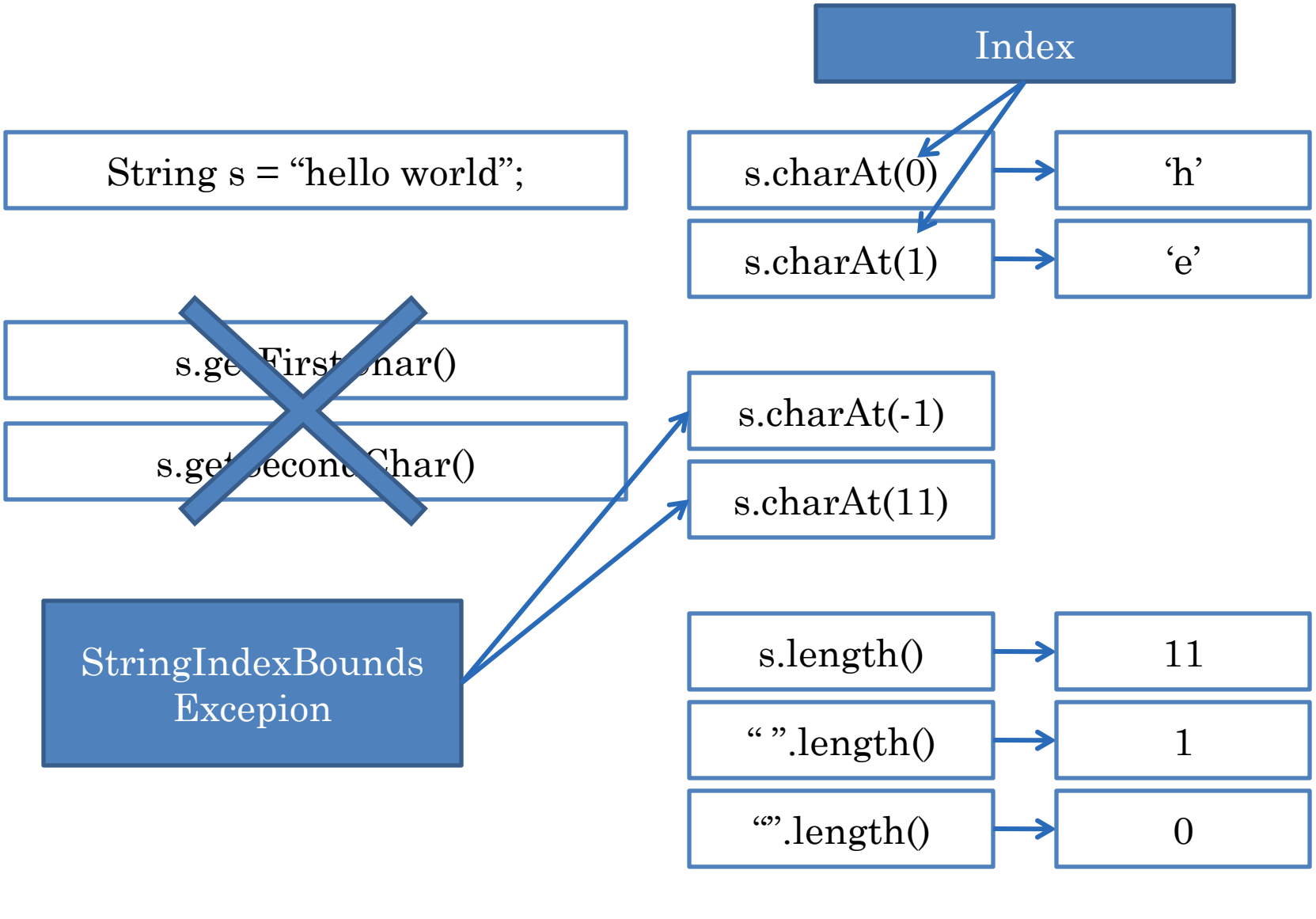
A Useful Character Operation



String Constants



Accessing String Components

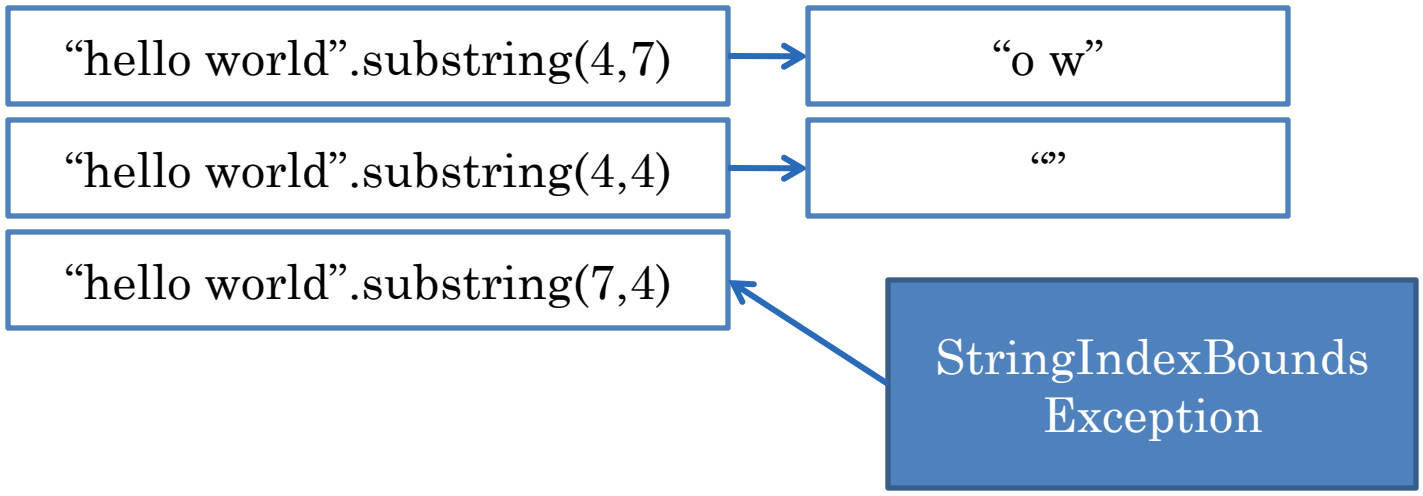


Accessing Substring

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

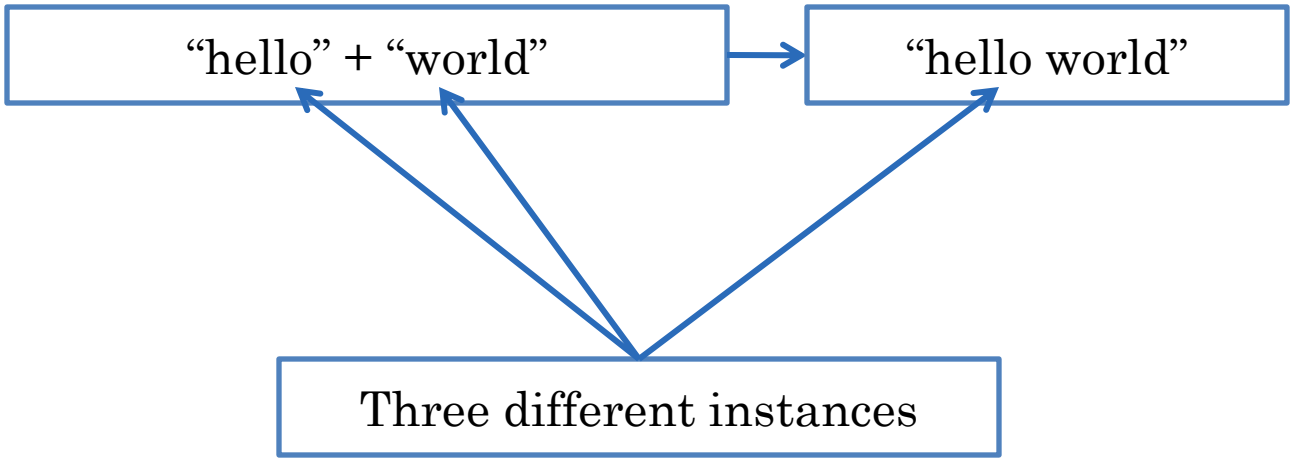
s.substring(beginIndex, endIndex)

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

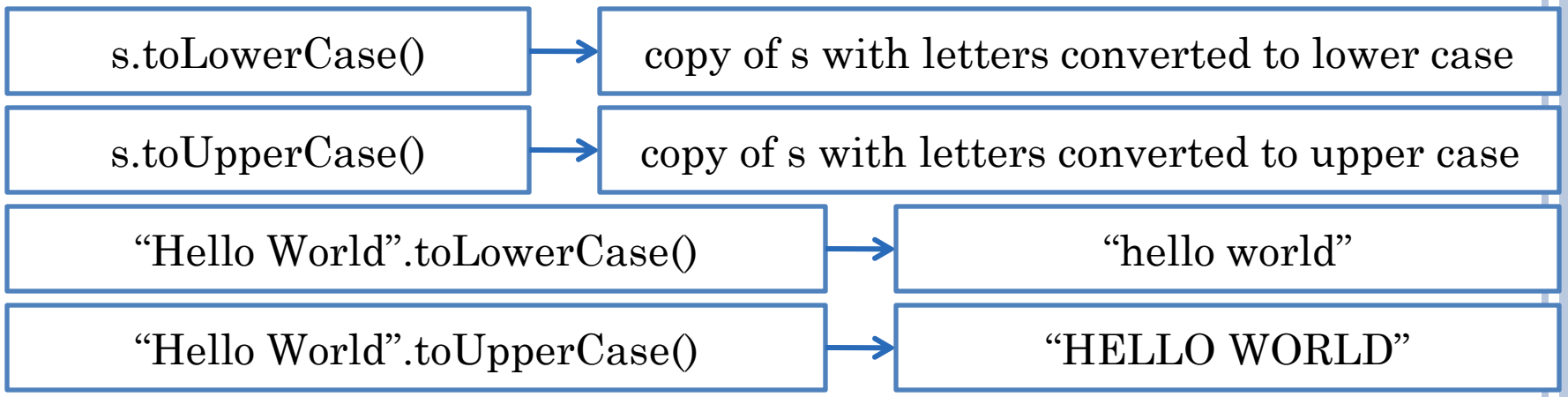


Changing Strings?

Stings 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

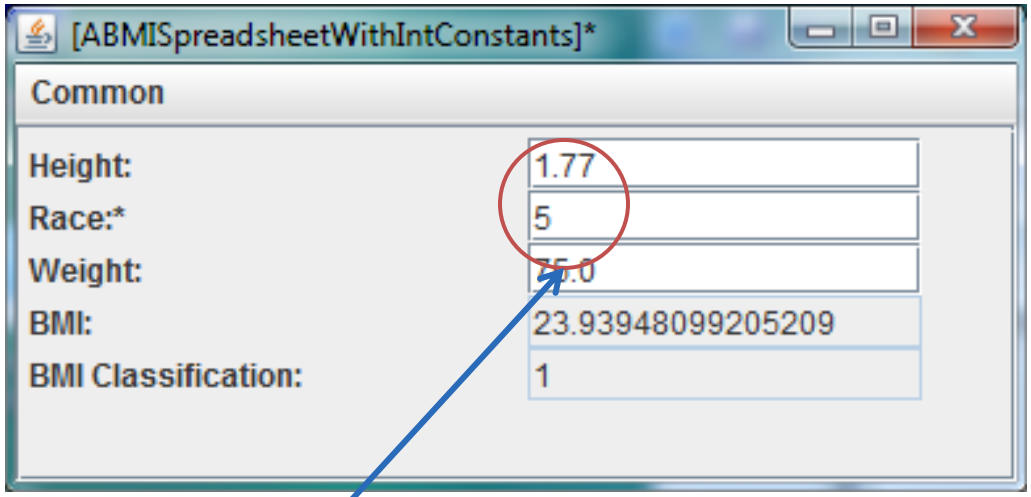
```
int race = IntRace.AFRICAN_AMERICAN;
public int getRace() {
  return race;
}
public void setRace (int newVal) {
  race = newVal;
}
public int getBMIClassification() {
  double bmi = getBMI();
  if (bmi <= MAX_UNDERWEIGHT_BMI)
    return BMIIntClassification.UNDER_WEIGHT;
  else if (bmi <= MAX_NORMAL_BMI)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



User can make mistake in assigning a value

String Constants

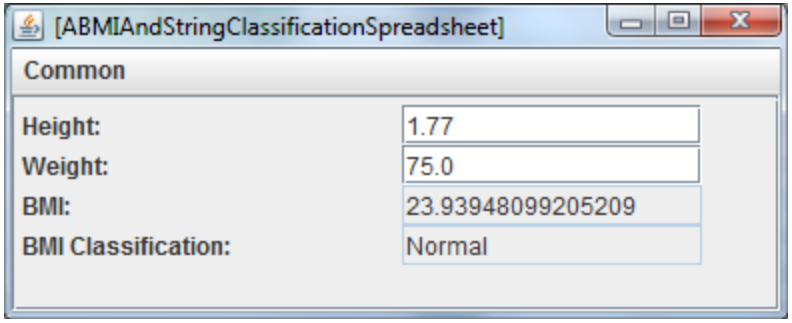
```
public interface StringBMIClassification {
  public static String UNDER_WEIGHT = "Under Weight";
  public static String NORMAL = "Normal";
  public static String OVER_WEIGHT = "Over Weight";
  public static String OBESE = "Obese";
  public static String EXTREMELY_OBESE = "Extremely Obese";
}
```

```
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;
}
```

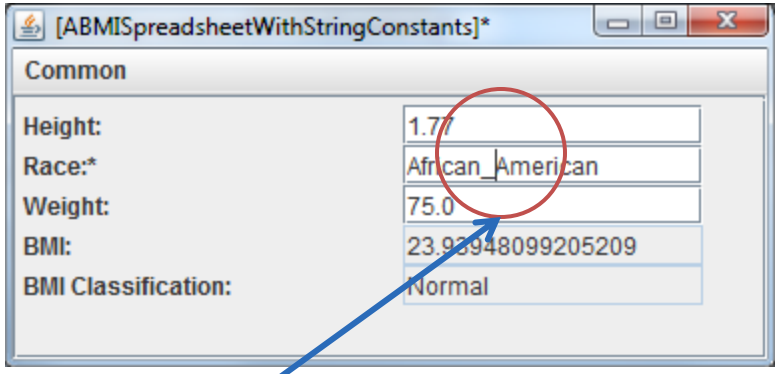
Errors Possible



String type is space inefficient

User or programmer can make mistake in assigning a value

String Issues



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 | EnumRace.java

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

EnumBMIClassification.java | 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