Université Concordia University
CENTRE FOR CONTINUING EDUCATION

INTRODUCTION TO PROGRAMMING WITH JAVA - CEJV416

Lecture #4

- String
- Math class
- Scanner, getting inputs

The String

- Strings consist of 0 or more characters.
- Can represent any sequence of characters
  - Moose
  - 42366X788
  - 5551212
  - The entire contents of a novel
- Dynamic
  - No fixed size
  - Is as large as required

The String

- Not a primitive
- String is a Java class
- The only class that supports the assignment operator =

```
String city = "Montreal";
city = "Toronto";
```

- If it isn't a number, true or false, or a single character* then its likely a String
- Strings can be empty

```
String city = "";
```

- Strings can be null

```
String city;
```

Going from number to String

- Two ways to convert a number to a String
- Use the wrapper class

```
String counterString = Integer.toString(counter);
String priceString = Double.toString(price);
```

- Use String concatenation with an empty String

```
String counterString = “”+counter;
String priceString = “”+price;
```

Going from String to number

- Possibility of failure exists
- Can only be done with wrapper classes

```
int quantity = Integer.parseInt(qtyString);
double price = Double.parseDouble(priceString);
```

- What happens if:

```
String myNumber = “234.67X”;
double distance = Double.parseDouble(myNumber);
```

The Evil Exception


```
Exception in thread "main" java.lang.NumberFormatException: For input string: "234.67X"
        at sun.misc.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:1241)
        at java.lang.Double.parseDouble(Double.java:540)
        at com.kenfogel.basicproject.BasicClassApp.perform(BasicClassApp.java:41)
        at com.kenfogel.basicproject.BasicClassApp.main(BasicClassApp.java:56)
Java Result: 1
```

Working with Strings

- How to append one string to another with the + operator

```
firstName = "Bob";      // firstName is Bob
lastName = "Smith";     // lastName is Smith
name = firstName + " "; // name is Bob followed by a space
name = name + lastName; // name is Bob Smith
```

- How to append one string to another with the += operator

```
firstName = "Bob";      // firstName is Bob
lastName = "Smith";     // lastName is Smith
name = firstName + " ";// name is Bob followed by a space
name += lastName;       // name is Bob Smith
```

Special characters in Strings
escape sequences

- Common escape sequences
  - **\n** newline
  - **\t** tab
  - **\r** return
  - **\"** double quotation mark
  - **** backslash

```
String text = “First line\nSecondLine”;
```

- will print out

```
First line
Second line
```

Console Output: System.out

- Two methods of the System.out object

```
println(data)
print(data)
```

- Code that uses the println method and prints a newline at the end

```
System.out.println(
    "Welcome to the Invoice Total Calculator");
System.out.println("Total: " + total);
System.out.println(message);
System.out.println();  // print a blank line
```

- Code that uses the print method

```
System.out.print("Total: ");
System.out.print(total);
System.out.print("\n");
```

Exercise 7

Using String

The Math class – java.lang.Math

- Math contains numerous math functions
  - `round(floatOrDouble)`
  - `pow(number, power)`
  - `sqrt(number)`
  - `max(a, b)`
  - `min(a, b)`
  - `random()`

Math examples

- The round method

```
long result = Math.round(1.667); // result is 2
int result = Math.round(1.49);   // result is 1
```

- The pow method

```
double result = Math.pow(2, 2);   // result is 4.0 (2*2)
double result = Math.pow(2, 3);   // result is 8.0 (2*2*2)
double result = Math.pow(5, 2);   // result is 25.0 (5 squared)
int result = (int)Math.pow(5, 2); // result is 25 (5 squared)
```

- The sqrt method

```
double result = Math.sqrt(20.25);   // result is 4.5
```

Math examples

- The max and min methods

```
int x = 67;
int y = 23;
int max = Math.max(x, y);     // max is 67
int min = Math.min(x, y);     // min is 23
```

- The random method

```
double x = Math.random() * 100;
            // result is a value >= 0.0 and < 100.0
long result = (long) x;
            // converts the result from double to long
```

Exercise 8

Math Problem

User Input – Round 1

- In Java all input is text
- Anything that is not text must be converted to its appropriate data type
- All GUI input is text
- A convenience class for console (non-GUI) input is available to assist in learning the fundamentals of Java without the complexity of a GUI
- This class is called Scanner

Scanner class – java.util.Scanner

- Accepts user input from the keyboard in a console application
- Has methods that accept user input and converts it to the appropriate data type
- Creating a Scanner object

```
Scanner sc = new Scanner(System.in);
```

Scanner class

- Common methods of a Scanner object

```
next()
nextInt()
nextDouble()
nextLine()
```

- How to use the methods of a Scanner object

```
String name = sc.next();
int count = sc.nextInt();
double subtotal = sc.nextDouble();
String cityName = sc.nextLine();
```

Scanner Example


```
Scanner s = new Scanner (System.in);

System.out.println("Please provide a double number");

double input = s.nextDouble();

System.out.println("User input is: " + input);
```

Don't forget to include:

```
import java.util.Scanner;
```

Exercise 9

Math Problems

Exercise for Home

- Revisit your tax calculation to get the price and tax values from the user

- Write a program to get the user's first name and last name of user and prints her initials