COP-2210
Computer Programming I

Instructor: Dr. Antonio Hernandez

Text: Big Java: Early Objects, Interactive Edition, 6th Edition

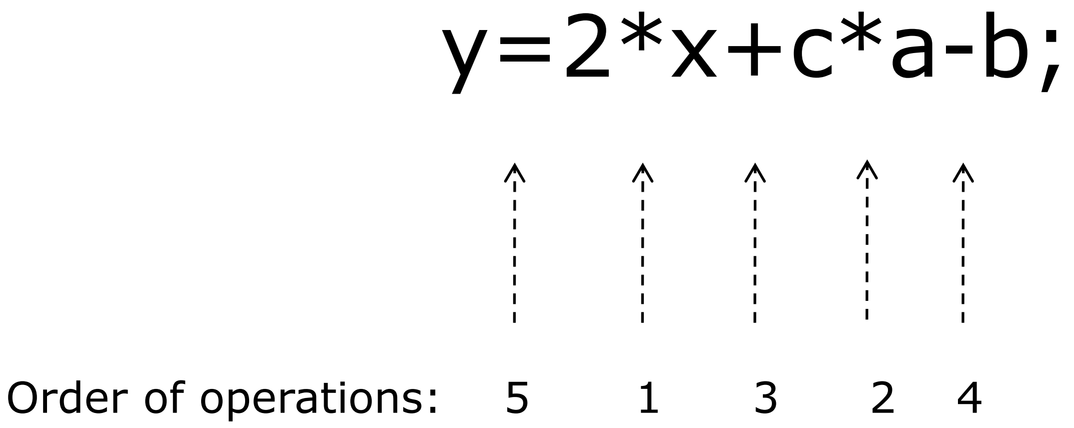
Operators

In math:

$$x + y * z \neq (x + y) * z$$

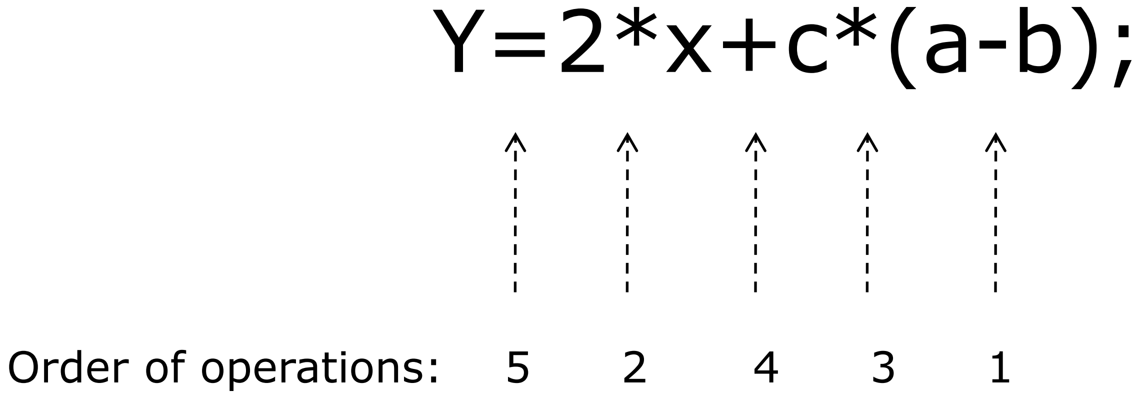
Precedence of operations: ***** , ÷ , **+** , **-**

Example: Precedence and Associativity of Operators



```
y=2*x+c*a-b;
```

Order of operations: 5 1 3 2 4



```
Y=2*x+c*(a-b);
```

Order of operations: 5 2 4 3 1

Precedence of Operators: *Try it yourself*

```
// Program 12_02: Celsius – Fahrenheit conversion

import java.util.*;

public class Prog12_02
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);

        double cTemp, fTemp;
        System.out.print("Enter a Celsius temperature: ");
        cTemp = in.nextDouble();

        fTemp =  9.0/5.0 * cTemp + 32.0;

        System.out.println ( "The Fahrenheit temp is " + fTemp + " degrees");
    }
}
```

Intermediate results with arithmetic operators

<operand$_1$>   <operator>   <operand$_2$>

<u>Rule</u>:

- If both operands have the same type, the result is of that type.
- If the two operands have different types, a *promotion operation* is performed (smaller data type → larger data type).

Pitfalls with arithmetic operations: *Example*

```
farTemp = 9.0/5.0*celsius + 32.0;   ←  Correct

farTemp = 9/5*celsius + 32.0;       ←  Incorrect!
```

Assignment operator

Rule:

- When numeric values are assigned to Java variables, “bigger” types cannot be assigned to “smaller” types (Java does this to avoid loss of precision). Doing so will result in a compilation error. So for example, *doubles* cannot be assigned to *floats* or *integers*.

Arithmetic Operations: Using Different Types

```
// Program 12_03: using different number types

public class Prog12_03
{
    public static void main ( String args[ ] )
    {

          int       i;
          float     f;
          double    d;

          int       i1 = 22,  i2 = 7;
          float     f1 = 22,  f2 = 7;
          double    d1 = 22, d2 = 7;
```

```
          i = i1 / i2;         System.out.println(i);

          f = i1 / f2;         System.out.println(f);

          f = f1 / f2;         System.out.println(f);

          d = i1 / d2;         System.out.println(d);

          d = f1 / f2;         System.out.println(d);

          d = f1 / d2;         System.out.println(d);

          d = d1 / d2;         System.out.println(d);
    }
}
```

Arithmetic Operations: Using Different Types - result

Output of Program 12_03:

```
3
3.142857
3.142857
3.142857142857143
3.142857074737549
3.142857142857143
3.142857142857143
```

Note the different decimals

Operators in Java: precedence of operations

To override the implicit precedence of operations:

Use *parenthesis*

Observations:

- When to use parenthesis?

If you are not sure, *use them*.

- Nested parenthesis: do not use [] or {}. Use

(….(….(..)….)….)

PRACTICE

Program 12_04:

Write a program to calculate the temperature in Celsius degrees, given the Fahrenheit temperature

Input: Fahr. Temp. **Output**: Celsius Temp.

$$\text{cTemp} = \frac{5}{9}(\text{fTemp} - 32)$$

PRACTICE - ANSWER

```
// Program 12_04: Fahrenheit - Celsius conversion

import java.util.*;

public class Prog12_04 {
   public static void main(String args[]) {
      Scanner in = new Scanner(System.in);

      double cTemp, fTemp;
      System.out.print("Enter a Fahrenheit temperature: ");
      fTemp = in.nextDouble();

      cTemp =  5.0/9.0 * (fTemp - 32.0);

      System.out.println ( "The Celsius temp is " + cTemp + " degrees");
   }
}
```

The ***Math*** class

Math class: contains methods for mathematical calculations

Math class: part of the package *java.lang* (automatically imported)

<u>Examples of methods</u>

int **abs** (int);
double **cos** (double);
double **exp** (double);
double **pow** (double, double);
double **sin** (double);
double **tan** (double);
double **sqrt** (double);

Using the *Math* class: *Example*
Calculating the volume of a sphere

```
// Program 12_05: using the Math class

import java.util.*;

public class Prog12_05 {
     public static void main ( String args [ ] ) {
             double sphereVolume, rad;

             Scanner cin = new Scanner (System.in);

             System.out.print ( "Enter the radius of the sphere: ");

             rad = cin.nextDouble();

             sphereVolume  =  4  *  Math.PI  *  Math.pow ( rad,  3 )  /  3;

             System.out.println ( "The vol. of the sphere is: "  +  sphereVolume );
     }
}
```

$$V_S = \frac{4\pi r^3}{3}$$

PRACTICE

Program 12_06:

Write a program to calculate the distance between two points in the plane, **(x_1, y_1)** and **(x_2, y_2)**.

$$\text{distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

PRACTICE - ANSWER

```java
//Program 12_06: calculate the distance between two points
import java.util.Scanner;

public class Prog12_06 {
  public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    System.out.print("Enter the first point: ");
    double x1 = in.nextDouble();
    double y1 = in.nextDouble();

    System.out.print("Enter the second point: ");
    double x2 = in.nextDouble();
    double y2 = in.nextDouble();

    double d = Math.sqrt(Math.pow(x1-x2, 2) + Math.pow(y1-y2, 2));
    System.out.println("The distance between the two points is " + d);
  }
}
```

The Java Language

13. Using Classes Already Defined

Declaration of a class variable

Declaration:

<class name> <variable name> = new <class name> (<args.>)

It could be empty

Examples

```
Scanner in  =  new Scanner ( System.in );
Point p  =  new Point ( 2, -1 );
Student s  =  new Student ( );
```

Declaration of a class variable

OOP Terminology:

1) *Object* = variable whose type is a class
2) An object is an *instance* of a class
3) *Instantiation*: creation (declaration) of an object

Note: In the object creation process, memory is allocated for the object, which is done by the new operator

Declaration of a class variable: constructors

Constructor: It is a special type of method of an object.

1) Java will execute whatever code is inside the constructor when the object is created
2) It is mainly used for initialization purposes
3) It is *invoked* in the declaration:

<u>Example</u>

```
BufferedReader br  =  new BufferedReader ( isr );
```

invoking the constructor

Constructors: the *DecimalFormat* class

DecimalFormat: class used to get numbers formatted in a number of ways. Contained in the *java.text* package.

DecimalFormat constructors	Explanation
DecimalFormat ( )	Creates a *DecimalFormat* object with a default pattern and default symbols
DecimalFormat (String pattern)	Creates a *DecimalFormat* object with a given pattern and default symbols
DecimalFormat (String pattern, DecimalFormatSymbols symbols)	Creates a *DecimalFormat* object with a given pattern and symbols

Access to *variables* and *methods* in a class

Access

Variables and methods in an object: may be accessed by

<object name> . <variable name>

<object name> . <method name>

Example

```
•
•
•

System.out.print ("Enter name: ");

s = in.nextLine ( );
```

Static methods and attributes

Static method ***(the declaration to be study later)***

Allows the programmer to use the class name to call the static method directly, as opposed to making an object and calling the method by using the object

Static attribute ***(the declaration to be study later)***

Allows the programmer to use the class name to access the static attribute directly, as opposed to making an object and accessing the attribute by using the object

Example

```
SphereVolume   =   4  *  Math.PI  *  Math.pow ( r,  3 )  /  3;

JOptionPane.showMessageDialog ( . . . );
```

Using the DecimalFormat class: Prelude

Try it yourself

```
//Prog13_01 - Using formulas with real numbers
import java.util.*;

public class Prog13_01
{
   public static void main(String args[])
   {
      Scanner input = new Scanner(System.in);

      System.out.print("Enter the radius of a circle: ");
      double radius = input.nextDouble();
      double area = Math.PI * Math.pow(radius, 2);

      System.out.println("Area = " + area);
   }
}
```

Using the DecimalFormat class: Prelude

Try it yourself

```
//Prog13_02 - Using  DecimalFormat class to format output
import java.util.*;
import java.text.*;

public class Prog13_02 {
   public static void main(String args[]) {
      Scanner input = new Scanner(System.in);

      System.out.print("Enter the radius of a circle: ");
      double radius = input.nextDouble();
      double area = Math.PI * Math.pow(radius, 2);

      DecimalFormat df  =  new  DecimalFormat ();
      df.setMaximumFractionDigits(2);

      System.out.println("Area = " + area);
      System.out.println("Area = " + df.format(area));    } }
```

PRACTICE

Program 13_03:

Redo Prog12_04 (calculate the temperature in Celsius degrees, given the Fahrenheit temperature) to display cTemp with 2 decimal places

Input: Fahr. Temp. **Output**: Celsius Temp.

$$\text{cTemp} = \frac{5}{9}(\text{fTemp} - 32)$$

PRACTICE - ANSWER

```
// Program 12_04: Fahrenheit - Celsius conversion
import java.util.*;
import java.text.*;

public class Prog13_03 {
   public static void main(String args[]) {
      Scanner in = new Scanner(System.in);

      double cTemp, fTemp;
      System.out.print("Enter a Fahrenheit temperature: ");
      fTemp = in.nextDouble();

      cTemp =  5.0/9.0 * (fTemp - 32.0);

      DecimalFormat df = new  DecimalFormat();
      df.setMaximumFractionDigits(2);

      System.out.println ( "The Celsius temp is " + df.format(cTemp) + " degrees");
   }
}
```

Formatting output

JDK 5.0 incorporated the *printf* method from the C library

System.out.printf(“ . . . %<character> . . . ”, <var>, . . .);

Examples:

```
System.out.printf( “%8.2f ”, x );

System.out.printf(“Hello, %s. Next year, you’ll be %d”, name, age+1);
```

Formatting output

Conversion character	Type
d	integer
s	string
f	Floating point

Another way:

```
String s = String.format ( “Hello, %s. Next year . . . );
```

Formatting output *Try it yourself*

```
//Program 13_04 - Formatting output with printf

import java.util.*;

public class Prog13_04
{
    public static void main ( String args[] )
    {
          Scanner in = new Scanner ( System.in );

          System.out.println ( "Enter your name, age and the value of PI: " );

          String name = in.nextLine ( );
          int age = in.nextInt ( );
          double pi = in.nextDouble ( );

          System.out.printf ( "\nHello %s, you are %d and PI = %5.2f\n", name, age, pi );
    }
}
```

Formatting output *Try it yourself*

```
//Program 13_05 - Formatting output with printf

import java.util.*;

public class Prog13_05
{
     public static void main( String args[] )
     {
               Scanner in = new Scanner ( System.in );

               System.out.println ( "Enter your name, age and the value of PI: " );

               String name = in.nextLine ( );
               int age = in.nextInt ( );
               double pi= in.nextDouble ( );

               String s = String.format ( "\nHello %s, you are %d and PI = %5.2f\n", name, age, pi );

               System.out.print ( s );
     }
}
```

String class methods: *charAt*

```
//Program 13_06 - Using String methods

import java.util.*;

public class Prog13_06
{
     public static void main( String args[] )
     {
          Scanner in = new Scanner(System.in);

          System.out.println("Do you want to end the program? (Yes/No)");

          String answer = in.nextLine();

          char c = answer.charAt(0);

          System.out.println("You entered '" + c + "'");

      }
}
```

String utilities

The String class contain several utilities that are useful in string manipulation. For example:

s.replaceFirst (s1, s2): replaces the first occurrence of substring s1 in s with the string given in s2

```
String s = "I LIKE practice and I LIKE theory";
String r = s.replaceFirst("LIKE", "LOVE");  // r = "I LOVE practice and I LIKE theory"
```

s.replaceAll (s1, s2): replaces all occurrences of substring s1 in s with the string given in s2

```
String s = "I LIKE practice and I LIKE theory";
String r = s.replaceAll("LIKE", "LOVE");  // r = "I LOVE practice and I LOVE theory"
```

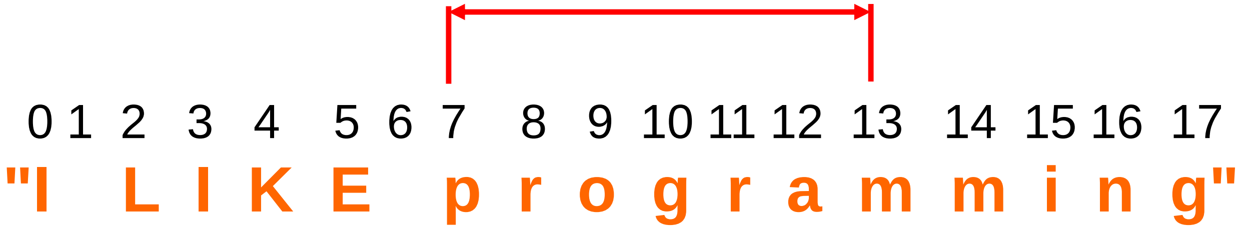
String utilities

s.length(): number of characters in s.

```
String s = "Hello, World!";
System.out.println(s.length()); // it displays 13
```

s.substring (loc1, loc2): returns a string that is a substring of s (the string that begins at location *loc1* and ends at location *loc2 - 1*).

```
String s = "I LIKE programming";
String r = s.substring(7, 14);  // r = "program"
```



0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

"I L I K E p r o g r a m m i n g"

String class methods: *replacing, finding substrings*

```
//Program 13_07 - Using String methods

import java.util.*;

public class Prog13_07
{
   public static void main( String args[] )
   {
         Scanner in = new Scanner(System.in);

         String s = "I LIKE programming in Java";
         System.out.println(s);

         System.out.print("\n (To continue, hit Enter)");
         in.nextLine();

         s = s.replaceAll("LIKE", "LOVE");
         System.out.println("\n" + s);
```

```
         System.out.print("\n (To continue, hit Enter)");
         in.nextLine();

         s = s.substring(0, 2) + "REALLY" + s.substring(1, s.length());
         System.out.println("\n" + s);

         System.out.print("\n (To continue, hit Enter)");
         in.nextLine();

         s = s.replaceAll("in Java", "LANGUAGES");
         System.out.println("\n" + s);
   }
}
```

replaceAll vs. replaceFirst

PRACTICE

Program 13_08:

Write a Java program that asks the user to enter a phrase with a word repeated twice in it and two different words to use as a replacement. An example of the program output:

```
Enter a phrase with 2 words in it repeated: Hello everyone. Hello!
Enter the word that is repeated: Hello
Enter the replacement for the first occurrence: Hi
Enter the replacement for the second occurrence: Good morning

New Phrase: Hi everyone. Good morning!
```