

Object Oriented Approach

Dr. youssef atat

Accessing Objects

- Referencing the object's data:
obj.data
- Invoking the object's method:
obj.method(arguments)

Constructor

- Constructors are a special kind of methods that are invoked to construct objects.
- A constructor with no parameters is referred to as a *no-arg constructor*.
- Constructors must have the same name as the class itself.
- Constructors do not have a return type—not even void.
- In Java, constructors are invoked using the new operator when an object is created. Constructors play the role of initializing objects.

Example

```
class Circle {  
    /** The radius of this circle */  
    double radius = 1.0; ← Data field  
  
    /** Construct a circle object */  
    Circle() {  
    }  
  
    /** Construct a circle object */  
    Circle(double newRadius) {  
        radius = newRadius;  
    }  
  
    /** Return the area of this circle */  
    double getArea() { ← Method  
        return radius * radius * 3.14159;  
    }  
}
```

The diagram illustrates the components of a Java class definition for a circle. It shows a class named `Circle` with three members:

- Data field:** A `double` variable named `radius` is initialized to `1.0`. This is annotated with a green arrow and the label "Data field".
- Constructors:** Two methods are defined to create `Circle` objects: a no-argument constructor `Circle()` and a parameterized constructor `Circle(double newRadius)` that sets the `radius` field. These are grouped by a bracket and annotated with a green arrow and the label "Constructors".
- Method:** A method named `getArea()` is defined to calculate the area of the circle using the formula `radius * radius * 3.14159`. This is annotated with a green arrow and the label "Method".

Example

```
ClassName objectRefVar = new ClassName();
```

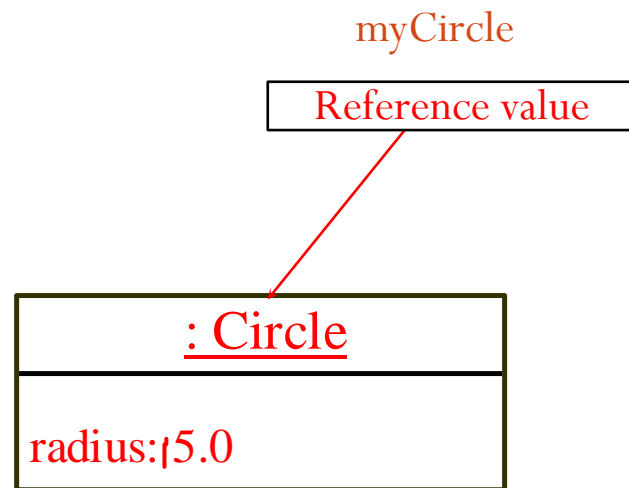
Example:

```
Circle myCircle = new Circle();
```

```
Circle myCircle1 = new Circle(5.0);
```

```
myCircle.getArea();
```

Example



JAVA

```
//This program prints Welcome to Java!  
  
public class Welcome {  
  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java!");  
    }  
}
```

JAVA

- **System.out.print / System.out.println**

```
System.out.println("Welcome to Java!");
```

- **int x=5;**

```
System.out.println("The value of x is "+x);
```

Numerical Data Types

<u>Name</u>	<u>Range</u>	<u>Storage Size</u>
byte	-2^7 (-128) to 2^7-1 (127)	8-bit signed
short	-2^{15} (-32768) to $2^{15}-1$ (32767)	16-bit signed
int	-2^{31} (-2147483648) to $2^{31}-1$ (2147483647)	32-bit signed
long	-2^{63} to $2^{63}-1$ (i.e., -9223372036854775808 to 9223372036854775807)	64-bit signed
float	Negative range: -3.4028235E+38 to -1.4E-45 Positive range: 1.4E-45 to 3.4028235E+38	32-bit IEEE 754
double	Negative range: -1.7976931348623157E+308 to -4.9E-324 Positive range: 4.9E-324 to 1.7976931348623157E+308	64-bit IEEE 754

Data types

- boolean : true or false
- Integer
- Float
- Double
- ...

Numeric Operators

Name	Meaning	Example	Result
+	Addition	34 + 1	35
-	Subtraction	34.0 - 0.1	33.9
*	Multiplication	300 * 30	9000
/	Division	1.0 / 2.0	0.5
%	Remainder	20 % 3	2

Increment and Decrement Operators

- `var++`
- `++var`
- `var--`
- `--var`

Shortcut Assignment Operators

Operator

Example Equivalent

`+=`

`i += 8`

`i = i + 8`

`--`

`f -= 8.0`

`f = f - 8.0`

`*=`

`i *= 8`

`i = i * 8`

`/=`

`i /= 8`

`i = i / 8`

`%=`

`i %= 8`

`i = i % 8`

String

- The char type only represents one character. To represent a string of characters, use the data type called String. For example,

```
String message = "Welcome to Java";
```

- String is a predefined class in the Java library
- String message = "Welcome " + "to " + "Java";//Concatenation

Getting Input Using Scanner

- `import java.util.Scanner;`
- Create a Scanner object

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

- Use the methods `next()`, `nextByte()`, `nextShort()`, `nextInt()`, `nextLong()`, `nextFloat()`, `nextDouble()`, or `nextBoolean()` to obtain to a string, byte, short, int, long, float, double, or boolean value. For example:

```
System.out.print("Enter a double value: ");  
Scanner scanner = new Scanner(System.in);  
double d = scanner.nextDouble();
```

Compile

- Compile: `javac Welcome.java`
 `Welcome.class`
- Run: `java Welcome`

Selection

- If else
- Nested if/else
- Comparison operators (<, <=, >,
- Boolean operators (&&, ||, !, ..)

Loops

- while
- for
- do while
- break
-

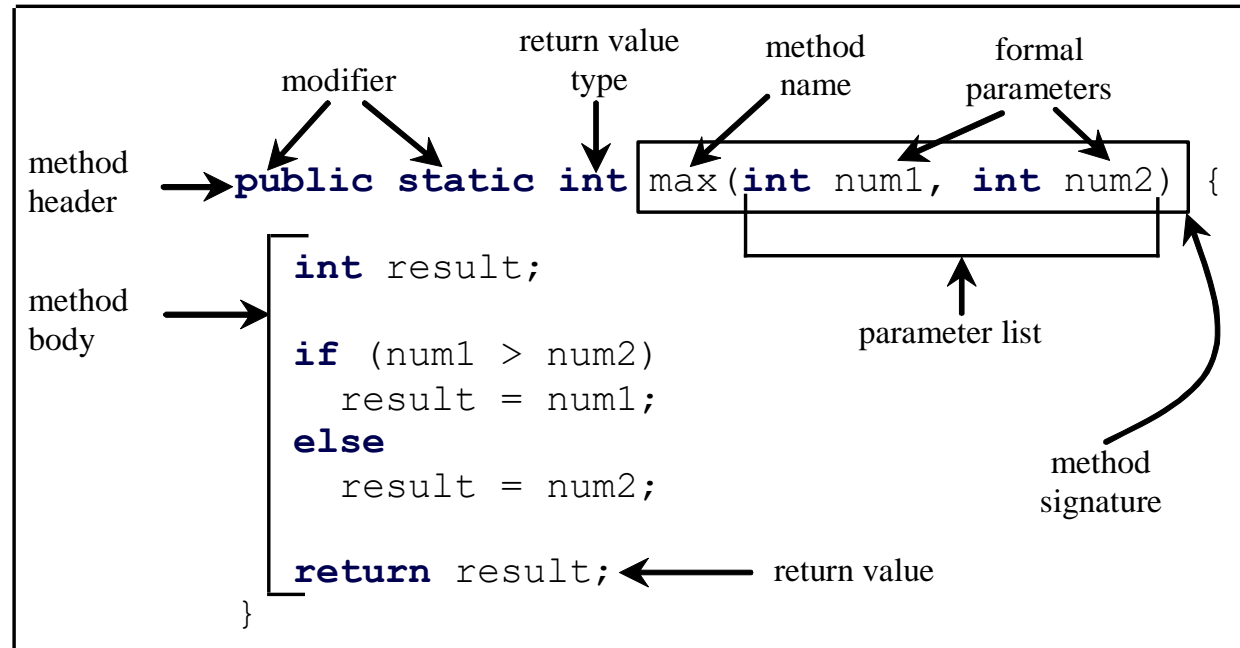
Array

- `int[] A = new int[10];`
- `int A[] = new int[10];` //not preferred

- `double[] mylist;`
 `mylist = new double[4];`
 or `double [] mylist = { 1.8, 2.5, 3.8, 10.2};`
- `length`: the length of an array.
 example: `mylist.length = 4`

Method

Define a method



Invoke a method

