BINUS UNIVERSITY

People
Innovation
Excellence

Course : Object Oriented Programming

Wrapper Class, Arithmetic Operation, Logic and Relational Operation

Session 2

BINUS UNIVERSITY

Introduction to Wrapper Class

- All primitive data types are wrapped into a class in Java and is fixed.
- Contains in java.lang *package*
- Used to symbolize the primitive data types in an object if necessary.
- Is the *final class* and *interface*.

People
Innovation
Excellence

Data Type to be Wrapped

- There are 8 primitive data type to be wrapped as Class in Java

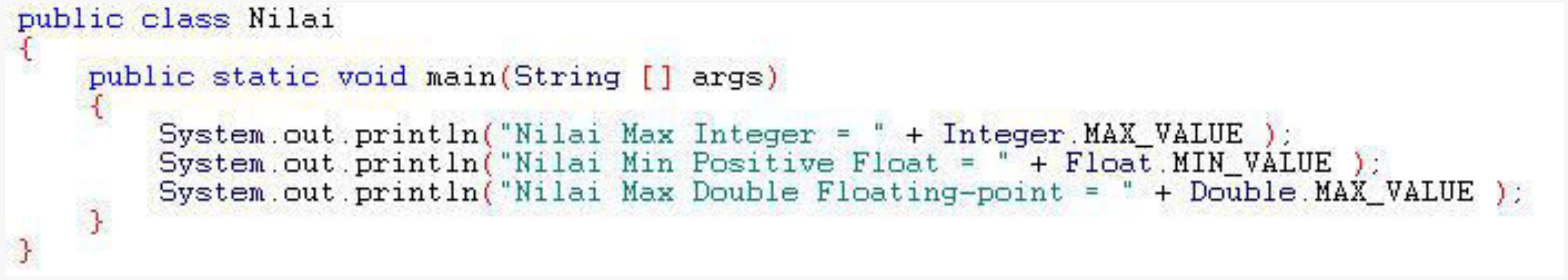
Instance of the *class* Number

Primitive type	Wrapper class	Constructor Arguments
byte	Byte	byte or String
short	Short	short or String
int	Integer	int or String
long	Long	long or String
float	Float	float, double or String
double	Double	double or String
char	Character	char
boolean	Boolean	boolean or String

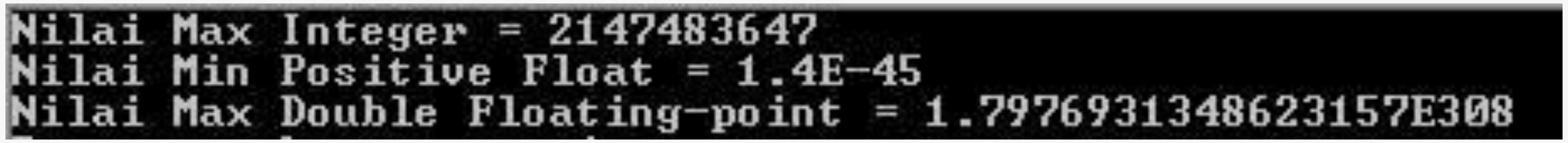
People Innovation Excellence

Wrapper Class Constants (Cont'd..)

- Example of use :



```
public class Nilai
{
    public static void main(String [] args)
    {
        System.out.println("Nilai Max Integer = " + Integer.MAX_VALUE );
        System.out.println("Nilai Min Positive Float = " + Float.MIN_VALUE );
        System.out.println("Nilai Max Double Floating-point = " + Double.MAX_VALUE );
    }
}
```



```
Nilai Max Integer = 2147483647
Nilai Min Positive Float = 1.4E-45
Nilai Max Double Floating-point = 1.7976931348623157E308
```

People
Innovation
Excellence

People
Innovation
Excellence

String

- A collection of some characters into an array (Array of Character)
- Declaration :

```
String msg = new String(“Welcome to Java”);
Atau
String msg = “Welcome to Java”;
```
← String Literal Object

- Also could be create from a collection of characters :

```
char[] charArray = {‘G’,’o’,’o’,’d’,’ ‘,’D’,’a’,’y’};
String msg = new String(charArray);
```

BINUS UNIVERSITY

String Method

- Functions of String Class:
  - length()
    - To find out how the length of the string
      - Example : msg.length();
  - charAt(index)
    - To restore the specific character designated by the index
      - Example : String msg ="Welcome";
        msg.charAt(0) ← then the result : W
  - concat()
    - To combine strings
      - Example: String word3 = word1.concat(word2);
        but we used to use String word3 = word1 + word2;
  - substring(start,finish)
    - To take a few characters from a string of the index.
      - Example : String msg = "Welcome to Java"
        msg.substring(0,6); ← then the result : Welcome
  - toLowerCase()
    - To convert all letters to lowercase
      - Example : "Welcome".toLowerCase(); ← then the result : welcome

People
Innovation
Excellence

People
Innovation
Excellence

Common used String Methods

Methods	Description
length()	Returns the length of this string.
charAt(index)	Returns the char value at the specified index.
concat()	Concatenates the specified string to the end of this string.
substring(start,finish)	Returns a new string that is a substring of this string.
toLowerCase()	Converts all of the characters in this String to lower case using the rules of the default locale.
toUpperCase()	Converts all of the characters in this String to upper case using the rules of the default locale.
trim()	Returns a copy of the string, with leading and trailing whitespace omitted.
replace(char oldChar, char newChar)	Returns a new string resulting from replacing all occurrences of oldChar in this string with newChar.
replaceFirst(String regex, String replacement)	Replaces the first substring of this string that matches the given regular expression with the given replacement.
split(String regex, int limit)	Splits this string around matches of the given regular expression.

People
Innovation
Excellence

Numeric Operators

Operator	Description	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

Numeric Operators

- **`%’** is an operator to calculate remainder/modulo from a division.
- This operator can be used with positive/negative number or decimal number.
- Example :

  10%7 = 3

  6 % 7 = 6

  -7 % 3 = -1

  -12 % 4 = 0

  20 % -13 = 7

  -26 % -8 = -2

People
Innovation
Excellence

People
Innovation
Excellence

Increment & Decrement Operator

- Shorthand operator increases or decreases 1 point.
- Usually use in looping.
- Operator: ++ and --
- It can be use as prefix which is before the variable or as postfix which is after the variable.
- It must not be separated by space. (++, NOT + +)

People
Innovation
Excellence

Increment & Decrement Operator

Operator	Name	Description
++var	preincrement	var is incremented by 1 first before its value is used.
var++	postincrement	var is incremented by 1 after its value has been used.
--var	predecrement	var is decremented by 1 first before its value is used.
var--	postdecrement	var is incremented by 1 after its value has been used.

People
Innovation
Excellence

Comparator operator

Operator	Description	Example	Result
<	less than	1 < 2	true
<=	less than or equal to	1 <= 2	true
>	greater than	1 > 2	false
>=	greater than or equal to	1 >= 2	false
==	equal to	1 == 2	false
!=	not equal to	1 != 2	true

BINUS UNIVERSITY

People
Innovation
Excellence

Boolean Operator

Operator	Name	Description
!	not	logical negation
&&	and	logical conjunction
||	or	logical disjunction
^	exclusive or	logical exclusion

People
Innovation
Excellence

NOT (!)

p	!p	Example
true	false	!(1>2) is true, because (1>2) is false
false	true	!(1>0) is false, because (1>0) is true

- Operator not (!) inverts the original value.
- true → false and false → true

People
Innovation
Excellence

AND (&&)

p1	p2	p1 && p2	Example
false	false	false	(2>3) && (5>5) is false Because both (2>3) and (5>5) are false
false	true	false	(2>3) && (6>5) is false Because (2>3) is false
true	false	false	(6>5) && (2>3) is false Because (2>3) is false
true	true	true	(3>2) && (5>=5) is true Because both (3>2) and (5>=5) are true

- AND Operator (&&) is true when all of its operands are true.
- If one of its operand is false, then AND is false.

People
Innovation
Excellence

OR (||)

p1	p2	p1 || p2	Example
false	false	false	(2>3) || (5>5) is false Because both (2>3) and (5>5) are false
false	true	true	(2>3) || (6>5) is true Because (6>5) is true
true	false	true	(6>5) || (2>3) is true Because (6>5) is true
true	true	true	(3>2) || (5>=5) is true Because both (3>2) and (5>=5) are true

- OR (||) Operator is true if one of every its operand is true.
- If all of its operands become false then OR is false.

XOR (^)

p1	p2	p1 ^ p2	Example
false	false	false	(2>3) ^ (5>5) is false Because both (2>3) and (5>5) are false
false	true	true	(2>3) ^ (6>5) is true Because (2>3) is false and (6>5) is true
true	false	true	(6>5) ^ (2>3) is true Because (6>5) is true and (2>3) is false
true	true	false	(3>2) ^ (5>=5) is true Because both (3>2) and (5>=5) are true

- Operator XOR (^) is true when both if its operands has different condition.
- When its operands has same condition then XOR is false.

People
Innovation
Excellence

Exception Handling

- 3 error type:
  - **Syntax errors** (compile errors) → against syntax rule of coding programming, founded at compile process by compiler
  - **Logic errors** (bug) → logic errors, obtain deviate output/performance
  - **Runtime errors** → false operation at program execution, program terminated
- Runtime errors : exception
- Exception cause terminate program
- Example:
  - Customer A transfer money to the account of customer B, when account had reduced and account B not yet increase, exception terminate program had happen. Customer A loses his money.

People
Innovation
Excellence

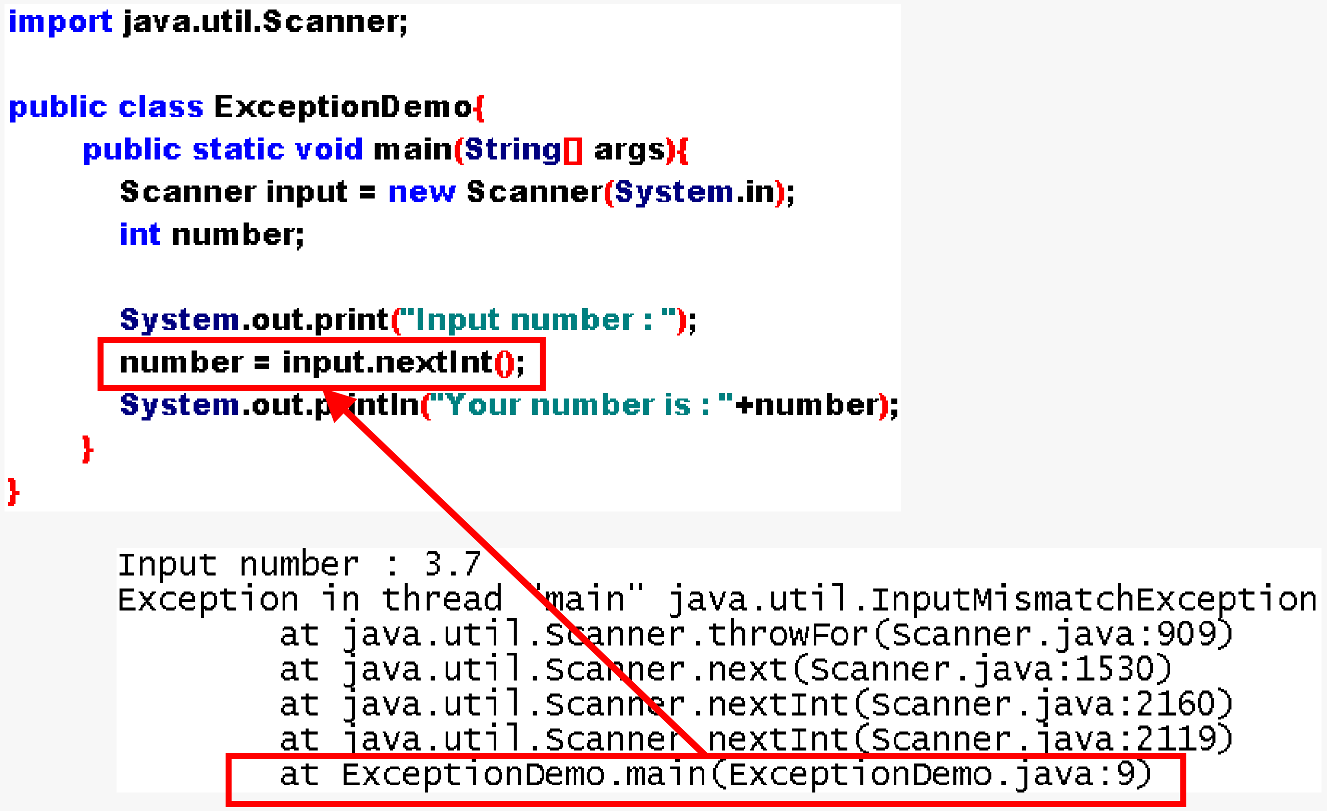
People
Innovation
Excellence

Exception Handling

- Handling runtime errors (exception handling)
- Using try and catch
- Type of error generally occur are:
  - In-correct input
  - Arithmetic (divided by 0)
  - Surpass array boundary that had been set
  - Un-initialized Object
- If error not handled / catch, than error will continued to the next handling
- Error not handled will cause program terminated.

People
Innovation
Excellence

Exception Handling Overview (1)

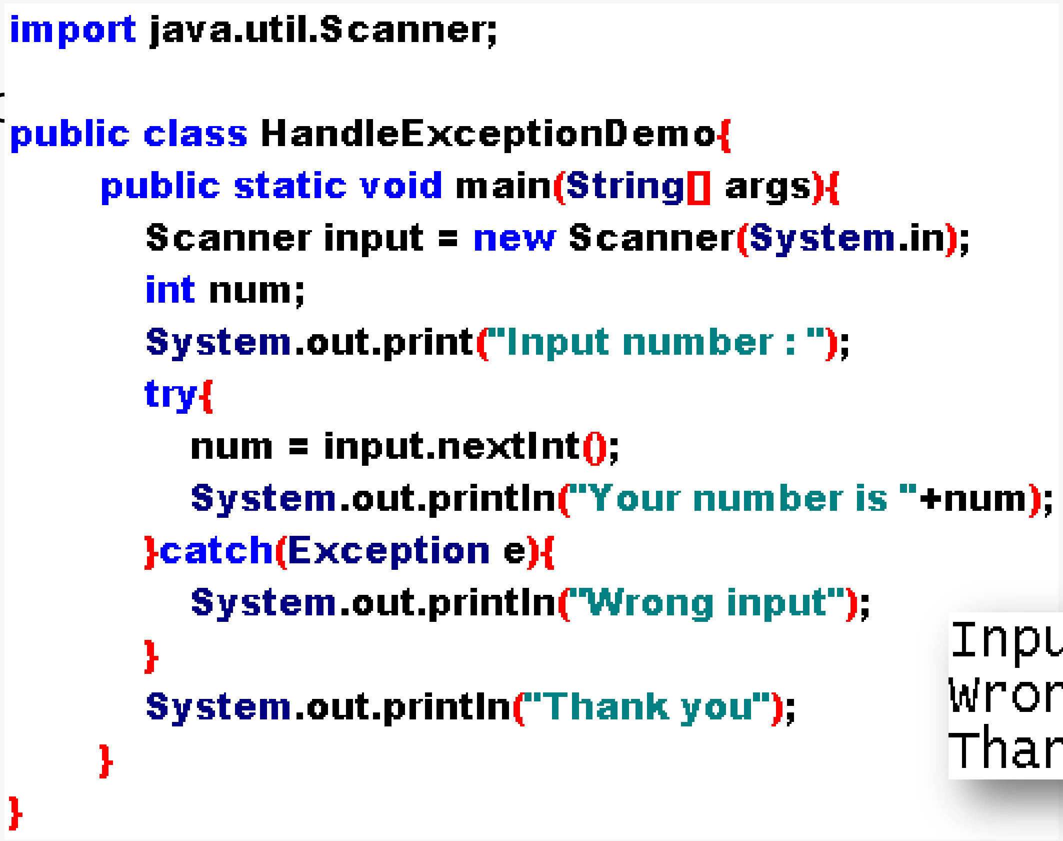


```java
import java.util.Scanner;

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

        System.out.print("Input number : ");
        number = input.nextInt();
        System.out.println("Your number is : "+number);
     }
}
```

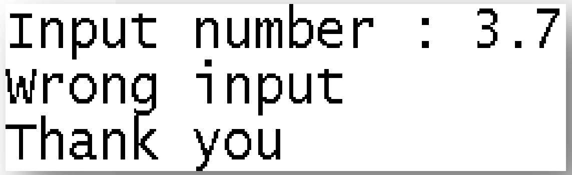
```
Input number : 3.7
Exception in thread "main" java.util.InputMismatchException
        at java.util.Scanner.throwFor(Scanner.java:909)
        at java.util.Scanner.next(Scanner.java:1530)
        at java.util.Scanner.nextInt(Scanner.java:2160)
        at java.util.Scanner.nextInt(Scanner.java:2119)
        at ExceptionDemo.main(ExceptionDemo.java:9)
```

Exception-Handling Overview (2)

- Sc



```
import java.util.Scanner;

public class HandleExceptionDemo{
    public static void main(String[] args){
      Scanner input = new Scanner(System.in);
      int num;
      System.out.print("Input number : ");
      try{
        num = input.nextInt();
        System.out.println("Your number is "+num);
      }catch(Exception e){
        System.out.println("Wrong input");
      }
      System.out.println("Thank you");
    }
}
```



```
Input number : 3.7
Wrong input
Thank you
```