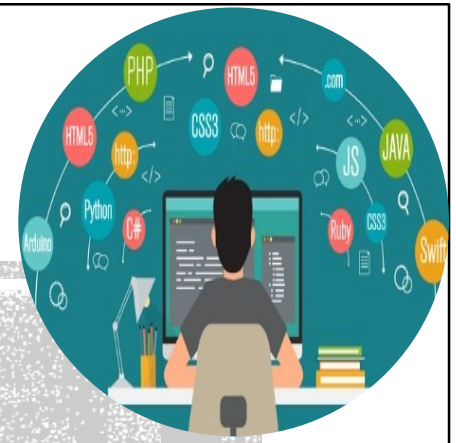
جامعة التنمية البشرية
UNIVERSITY OF HUMAN DEVELOPMENT



COMPUTER PROGRAMMING CONCEPTS

CS&IT 1101
Instructor: Shakar H. Salih
E-mail: shakar.salih@uhd.edu.iq

LECTURE 4 :BASIC ELEMENTS OF JAVA

OUTLINES

- Type conversion(Casting)
- String
- String and operator +
- Text codes
  1. Ascii
  2. Unicode
- Type of Error
- Input Read Statement
- Increment and decrement operator
- Escape character

TYPE CONVERSION (CASTING)

- byte → short → int → long (implicit).
- Implicit type coercion: is the process of converting value from one type to another.
- byte ← short ← int ← long (explicit casting).
- Cast operator: is the process of converting one data type to another.

Syntax:

(dataTypeName) expression

NOTE

1. First, the expression is evaluated. Its value is then treated as a value of the type specified by dataTypeName.
2. When using the cast operator to treat a floating-point (decimal) number as an integer, you simply drop the decimal part of the floating-point number. That is, the floating-point number is cutting off.
3. You can also use cast operators to explicitly treat char data values as int data values, and int data values as char data values. To treat char data values as int data values, you use a collating sequence.

For example, in the Unicode character set, (int)('A') is 65 and (int)('8') is 56. Similarly, (char)(65) is 'A' and (char)(56) is '8'.

EXAMPLE (1/2)

Expression	Evaluates to
1. (int) (7.9)	7
2. (int) (3.3)	3
3. (double) (25)	25.0
4. (double) (5+3)	= (double) (8) = 8.0
5. (double) (15) /2	=15.0 / 2 (because (double) (15)=15.0) =15.0 / 2.0 =7.5

EXAMPLE(2/2)

6. (double) (15/2) =(double) (7) (because 15 / 2 =7)
   = 7.0

7. (int)(7.8+(double)(15)/2) = (int)(7.8 + 7.5)
   = (int) (15.3)
   = 15

8. (int)(7.8+(double)(15/2)) =(int)(7.8 + 7.0)
   =(int)(14.8)
   = 14

STRING

What about data values such as a person's name?

- A person's name contains more than one character. Such values are called strings.
- **String:** Is a sequence of zero or more characters. Strings in Java are enclosed in double quotation marks " string".
- A string that contains no characters is called a **null** or **empty** string. The following are examples of strings. Note that " " is the empty string.
  - ✓"Shakar Hussein"
  - ✓ "Ara"
  - ✓ " "
- Every character in a string has a specific position in the string. The position of the first character is 0, the position of the second character is 1, and so on. The **length** of a string is the number of characters in it.

EXAMPLE

String	"Sunny Day"								
Character in the string	'S'	'u'	'n'	'n'	'y'	' '	'D'	'a'	'y'
Position of the character in the string	0	1	2	3	4	5	6	7	8

The length of the string "Sunny Day" is 9

STRING AND THE OPERATOR +

- One of the most common operations performed on strings is the concatenation operation, which allows a string to be appended at the end of another string.
- The operator + can be used to concatenate (or join) two strings as well as a string and a numeric value or a character.
- "Sunny" + " Day“ This expression evaluates to "Sunny Day"

EXAMPLE

1. "The sum = " + 12 + 26

This expression evaluates to

"The sum = 1226“

2. "The sum = " + 12 + 26

is evaluated as follows:

"The sum = " + 12 + 26 = ("The sum = " + 12) + 26

= "The sum = 12" + 26

= "The sum = 1226"

EXAMPLE

3. "The sum = " + (12 + 26)

This expression evaluates as follows:

"The sum = " + (12 + 16)

= "The sum = " + 38

= "The sum = 38"

4. 12 + 26 + " is the sum"

This expression evaluates as follows:

12 + 26 + " is the sum"

= (12 + 26) + " is the sum“

= 38 + " is the sum"

= "38 is the sum"

EXAMPLE

5. "The sum of " + 12 + " and " + 26 + " = " + (12 + 26)

Notice the parentheses around 12 + 26.

This expression evaluates as follows:

"The sum of " + 12 + " and " + 26 + " = " + (12 + 26)
= "The sum of 12" + " and " + 26 + " = " + (12 + 26)
= "The sum of 12 and " + 26 + " = " + (12 + 26)
= "The sum of 12 and 26" + " = " + (12 + 26)
= "The sum of 12 and 26 = " + (12 + 26)
= "The sum of 12 and 26 = " + 38
= "The sum of 12 and 26 = 38"

TEXT CODES

- Computer programmers realized that they need a commonly agreed standard code in which numbers stood for alphabets, special characters and digits.
- EBCDIC, ASCII and Unicode are three most popular coding systems.

ASCII

- Every letter, number, or special symbol (such as * or {) on your keyboard is encoded as a sequence of bits, each having a unique representation.
- The most commonly used encoding scheme on personal computers is the seven-bit **American Standard Code for Information Interchange** (ASCII). The ASCII data set consists of 128 characters, numbered 0 through 127. (Note that 2^7= 128 and 2^8= 256.) That is, in the ASCII data set, the position of the first character is 0, the position of the second character is 1, and so on.

ASCII

- The characters from 0 to 31 are control characters.
- 32 to 64 special characters
- 65 to 96 uppercase letters and few symbols
- 97 to 127 lowercase letters and other symbols
- And 128 to 255 are other symbols.

UNICODE

- An developing standard is Unicode Worldwide Character Standard.
- It has two bytes 16 bits to represent each symbol.
- It can represent 65536 symbols, which include international symbols, and alphabets from different languages in the world.

NOTE

- The number system that we use in our daily life is called the decimal system or base 10. Because everything inside a computer is represented as a sequence of 0s and 1s, that is, binary numbers, the number system that a computer uses is called binary or base 2.
- The ASCII character set is a subset of Unicode; the first 128 characters of Unicode are the same as the characters in ASCII. If you are dealing with only the English language, the ASCII character set is sufficient to write Java programs.
- The advantage of the Unicode character set is that symbols from languages other than English can be handled easily.

TYPE OF ERRORS

- **What is the difference???**

Errors

- Something that make a program go wrong.
- Can give unexpected result.
- Types:
  1. Compile-time errors
  2. Run-time errors

Exception

- Condition that is caused by a run-rime error in the program.
- Ex. Dividing by zero.
- Interpreter creates an exception object and throws it.

COMPILE-TIME ERROR (1/2)

- Occurs at the time of compilation.
- Detected and displayed by the interpreter.
- .class file will not be created //compile error
- For successful compilation it need to be fixed.
- For ex. Missing semicolon or missing brackets. // Syntax errors

COMPILE-TIME ERROR (2/2)

- Misspelling of identifier or keyword
- Missing double quotes in string
- Use of undeclared variable
- Use of = in place of == operator
- Path not found
- Changing the value of variable which is declared final
- Word not found. // Lexical Error

RUN-TIME ERROR (1/2)

- Program compile successfully
- Class file also generated.
- Though may not run successfully
- Or may produce wrong o/p due to wrong logic
- Error message generated
- For ex. Divide by zero. // logical Error

RUN-TIME ERROR (2/2)

- Accessing an element that is out of bound of an array. // aray run time
- Trying to store a value into an array of an incompatible class or type.
- Passing a parameter that is not in a valid range.
- Attempting to use a negative size for an array.
- Converting invalid string to a number
- Accessing a character that is out of bound of a string.
- Put wrong equation // Semantic Error

INPUT (READ) STATEMENT

- You've already seen that output can be displayed to the user using the subroutine **System.out.print**. This subroutine is part of a pre-defined object called **System.out**. The purpose of this object is precisely to display output to the user.
- There is also a corresponding object called **System.in** that exists to read data input by the user.
- To use System.in, a Java Scanner is used that will get the inputs.

SCANNER CREATION/DECLARATION

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

- Scanner class is one of the class of Java which provides the methods to get inputs.
- **System.in** is the parameter passed to the Scanner constructor so that Java will know to connect the new Scanner to the input. input is a reference that will store the location of newly created Scanner object.

SCANNER IMPORTS

- Scanner class is present in **java.util** package so we import this package in our program.
- In order to use Scanner, you must

```
Import java.util.Scanner;
```

SCANNER METHODS

NAME	USE
nextInt();	Returns the next integer value
nextDouble();	Returns the next double value
nextFloat();	Returns the next float value
nextLong();	Returns the next long value
nextShort();	Returns the next short value
next();	Returns the next one word String value
nextLine();	Returns the next multiple word String value

READING IN INTEGERS

We first create an object of Scanner class and then we use the methods of Scanner class.

```
Scanner keyboard = new Scanner(System.in);
System.out.print(" Enter Value");
int num = keyboard.nextInt();
```

- The nextInt() method is used to tell a Scanner object to retrieve the next integer value entered.
- In the example, the next integer typed in on the keyboard would be read in and placed in the integer variable num.
- nextInt() will read up to the first whitespace value entered.

INCREMENT AND DECREMENT OPERATOR(1/3)

- Statements are frequently used to keep track of how many times certain things have happened.
- The ++ is increment operator.
- The -- is decrement operator.
- Java provides the increment operator, ++, which increases the value of a variable by 1,
- The decrement operator, --, which decreases the value of a variable by 1
- Suppose **count** is an **int** variable. The statement:

```
count = count + 1;
```

INCREMENT AND DECREMENT OPERATOR(2/3)

- Increment and decrement operators each have two forms: **pre** and **post**.
- The syntax of the increment operator is:

Pre-increment:	**++variable**
Post-increment:	**variable++**

- The syntax of the decrement operator is:

Pre-decrement:	**--variable**
Post-decrement:	**variable--**

INCREMENT AND DECREMENT OPERATOR(3/3)

- What is the difference between the pre and post forms of these operators?
- The difference becomes apparent when the variable using these operators is employed in an expression.
- Suppose that x is a variable of type **int**. If ++x is used in an expression, first the value of x is incremented by 1, and then the new value of x is used to evaluate the expression. On the other hand, if x++ is used in an expression, first the current value of x is used in the expression, and then the value of x is incremented by 1.

EXAMPLE 1

```
int x=5;
int y=++x;
```

- The first statement assigns the value **5** to **x**. To evaluate the second statement, which uses the pre-increment operator, first the value of **x** is incremented to **6**, and then this value, **6**, is assigned to **y**. After the second statement executes, both **x** and **y** have the value **6**.

EXAMPLE 2

```
int x = 5;
int y = x++;
```

- As before, the first statement assigns **5** to **x**. In the second statement, the post-increment operator is applied to **x.** To execute the second statement, first the value of **x**, which is **5**, is used to evaluate the expression, and then the value of **x** is incremented to **6**. Finally, the value of the expression, which is **5**, is stored in **y**. After the second statement executes, the value of **x** is **6**, and the value of **y** is **5**.

EXAMPLE 3

1.
```
int a=5;
int b=2 + (++a);
```

2.
```
int a=5;
int b=2 + (a++);
```

ESCAPE CHARACTER

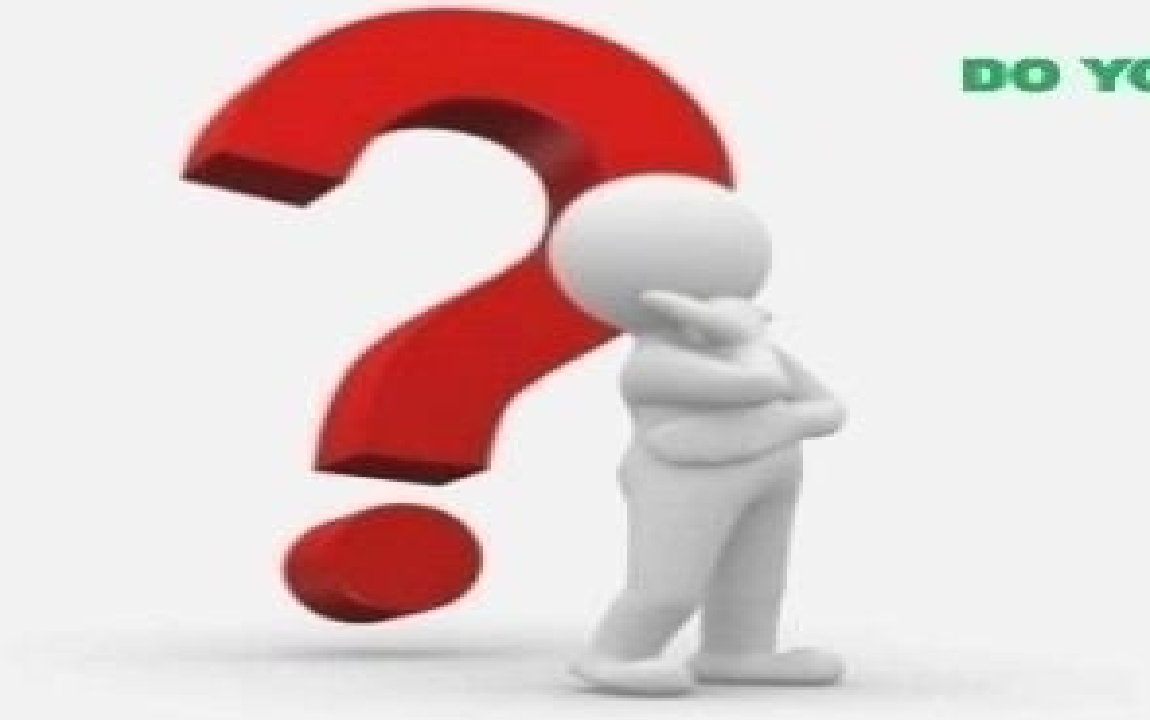
	Escape	Sequence Description
\n	Newline	Cursor moves to the beginning of the next line
\t	Tab	Cursor moves to the next tab stop
\b	Backspace	Cursor moves one space to the left
\r	Return	Cursor moves to the beginning of the current line (not the next line)
\\	Backslash	Backslash is printed
\’	Single quotation	Single quotation mark is printed
\”	Double quotation	double quotation mark is printed

REVIEW QUESTION

1. The operands of the modulus operator must be integers.
2. If the value of a is 4 and the value of b is 3, then after the statement a = b; the value of b is still 3.
3. Suppose x = 5. After the statement y = x++; executes, y is 5 and x is 6
4. Which of the following is a reserved word in Java?

   a. int b. INT c. Char d. CHAR

THANK YOU.....



DO YOU HAVE ANY QUESTIONS ?