University of Duhok
College of Science

Computer Skills

Part 2

Sarwar Mohammed Ali
email: sarwar.mustafa@uod.ac

2021-2022

Introduction to New Technologies of Computer

1. Artificial Intelligence (AI):

AI is being used everywhere starting from smartphones to cars and various other electronic gadgets. It has been a recent trend in technology and the world cannot survive without this.

2. Blockchain

This technology produces virtual currency, bitcoin that hit the market a lot. The currency, bitcoin has taken over the whole world with an increasing currency rate.

3. Augmented Reality (AR) and Virtual Reality (VR)

Both AR and VR technologies are fast emerging one which lets everyone experience virtually, that are very close to real. There has been significant growth in gaming and AR and VR gadgets from past years. Various big business enterprise solutions take advantage of its users such as in 3D projection, motion gesture making it more interactive and futuristic.

4. Deep Learning (DL)

Deep Learning, which is based on machine learning, is structured learning based on artificial neural networks. Deep Learning uses multiple layers to extract higher-level output from the raw input. In image processing, edges are lower layers while faces, digits that are the concepts of a human being considered as a higher-level.

6. Internet of Things (IoT)

IoT continues to be the most widely adopted use case for interrelated computing devices, digital machines, objects that transmits data without the requirement of human-to-human or human to computer interaction. It creates a virtual network by connecting various devices that work seamlessly through a single monitoring center. All the devices gather and share data about how they are used and the environments how they operate

7. Cybersecurity

It is the modern-day security feature that helps in protecting internet-connected systems including hardware, software and another security breach.

8. Big Data

Big data refers to those that are responsible for accessing and storing huge chunks of data. Most of the modern-day companies rely upon the big data to get hands-on customer base, product-related data, marketing research and a lot more.

9. RPA (Robotic Process Automation)

Robotic Process Automation allows everyone to automate daily routine and repetitive tasks. An industry that requires repetitive tasks or processes to continue, with the help of RPA, everything can be automated and there is no requirement of writing complex codes to automate such tasks.

Introduction to Windows Operators

Classification of Windows Operators

Windows Operators are broadly classified into three types. This classification is done based on the number of variables or operands an operator requires. The three types are:

- Unary Operators
- Binary Operators
- Ternary Operators

1. Unary Operators: They require a single operand

E.g. Prefix and Postfix operators, Shorthand operators, Negation Operator etc

2. Binary Operators: They require two operands to calculate the result.

E.g. Arithmetic operators, logical operators etc.

3. Ternary Operators: They require three operands.

E.g. Ternary Conditional operator

The various types of windows operators, based on their :functionality are

1. Basic Arithmetic Operators
2. Assignment Operator (=)
3. Comparison Operators
4. Prefix and Postfix Operators
5. Shorthand Operators
6. Logical Operators
7. Bitwise Operators
8. Ternary Operator
9. Operator Precedence

1. Basic Arithmetic Operators

These windows operators perform mathematical calculations.

Plus operator (+): Adds or concatenates the two operands.

E.g.

Sum of two integers: 1+3 results in 4

Sum of two floating point numbers: 9.8+0.4 results in 10.2

Concatenation of two strings: “Hello”+”World” results in “Hello World”

Minus Operator (-): Subtracts the second operand from first. Doesn't work on strings.

E.g.

Subtraction of two integers: 5-4 results in 1

Subtraction of two floating point numbers: 4.1 – 4.6 results in -0.5

Multiplication Operator (*): Multiplies the two operands.

E.g.

Multiplication of two integers: 9*5 results in 45

Multiplication of two floating point numbers: 1.1*2.3 results in 2.53

Division Operator (/): Divides the first operand by the second and returns the quotient as the result. The remainder is discarded. Some advanced languages, however, do not discard the remainder and keep dividing until a pre-set number of precision points is reached.

E.g.

Division of two integers: 45/11 results in 4

In advanced languages: 45/11 results in 4.090909

Modulus Operator (%): Divides the first operand by the second and returns the remainder as the result. The quotient is discarded. Doesn't work on floating point numbers.

E.g.

Modulus of two integers: 45/11 results in 1

2. Assignment Operator (=)

Assigns the result calculated in the right-hand side of the operator (RHS) to the left-hand variable (LHS). The left of the operator should always be a variable and not a constant/expression.

E.g.

- x = 5, assigns value 5 to x.
- 5 = x is invalid as left-hand side is a constant.
- y = x*4 calculates x*4 and assigns to y. Thus, y now holds the value 20
- x*4 = y is invalid as the left-hand side is an expression.

3. Comparison Operators

They compare the value of the first operand with that of the second operand and returns either true or false. These are less than (<), greater than (>), less than or equal (<=), greater than or equal (>=), equal (==), not equal (!=).

E.g.

- 61>45, returns true.
- 3==3, returns true.

4. Prefix and Postfix Operators

These windows operators increment or decrement the value of an operand by 1. They work only on integers.

E.g.

- **x=5**
- x++, x is now 6
- x, x is now 5 again

There is a very significant difference in the functioning of the two operators. Prefix operators change the value of the operand before evaluating the expression, whereas the postfix operator changes the value after the expression has been evaluated.

- **x = 5**

  print(x++), this will print 5 and then change the value of x to 6

  print(++x), this will increment the value from 6 to 7 and then print 7.

5. Shorthand Operators

These windows operators are a combination of two operators. The result is calculated using the existing value of the operand and assigned back to itself. They help minimize the lines of code written. The most common shorthand operators are:

+=: This is equivalent to addition and assignment.

-=: This is equivalent to subtraction and assignment.

***=:** This is equivalent to multiplication and assignment.

/=: This is equivalent to division and assignment.

E.g. x+=5, is equivalent to x=x+5.

6. Logical Operators

Logical operators are mainly used to control the program flow. Usually, they help the compiler on which path to follow based on the outcome of a decision. They always result in Boolean values

Logical AND (&&): Returns true if the conditions on both left and right side of the operator are true, otherwise returns false.

E.g.

- (2>3) && (4<5) returns false. Reason, 2 is not greater than 3
- Boolean b1 = true
  Boolean b2 = true
  b1 && b2 returns true.

Logical OR (||): Returns true if any of the operands is true, otherwise returns false.

E.g.

- (2>3) || (4<5) returns true
- Boolean b1 = false
  Boolean b2 = false
  b1 || b2 returns false.

Logical NOT / Negation (!): Inverses the result of the operand i.e. true becomes false and false becomes true.

E.g.

- ! (2>3) returns true
- ! (2>3) && (4<5) returns true. Reason -! (2>3) results in true.

7. Bitwise Operators

Bitwise operators are a special category of operators as they do not operate in a conventional way. While all other operators operate on bytes, bitwise operators operate on bits.

E.g.

Let us assume we have two numbers 2 and 4. Their respective binary conversions would be 0010 and 0100. Since 1 byte contains 8 bits, we convert them to 0000 0010 and 0000 0010.

- **Bitwise AND (&):** 2 & 4 results in 0000 0000 which is simply 0
- **Bitwise OR (|):** 2 | 4 results in 0000 0110 which is 6
- **Bitwise NOT (~):** ~2 results in 1111 1101 which is

8. Ternary Operator

The ternary operator is a shorthand operator for a logical if and else program flow. It evaluates the expression on the left of the question mark (?) and based on the result (true/ false) the operations on the left and right of the colon (:) are performed.

E.g.

(condition)? (operation if true): (**operation if false**)

(5>9) ? (print true): (print false) false is printed.

9. Operator Precedence

The precedence of operators is as follows (highest to lowest priority):

- Brackets
- Prefix and Postfix operators
- Multiplication, Division, Modulus
- Addition, Subtraction
- Bitwise Operators
- Logical Operators (some logical operators take higher precedence than bitwise operators. Learn more when you deep dive in bitwise operator section.)
- Ternary Operator
- Assignment, Shorthand operators