**WEEK 1- JAVA ASSIGNMENT**

**-BY ANKUR CHAUDHARY**

Ques no.1- What are the different types of comment symbols in java?

Answer no.1-

* /\* \*/ **Multiline Comment:** This type of comment must begin with **/\*** and end with **\*/**. Anything between these two comment symbols is ignored by the compiler.
* // **Single Line Comment:** A *single-line comment* begins with  
  a **//** and ends at the end of the line. As a general rule, programmers use multiline comments  
  for longer remarks and single-line comments for brief, line-by-line descriptions.
* /\*\*\*……..\*/  **Documentation Comment:** This type of comment  
  is used to produce an HTML file that documents your program. Thedocumentation comment begins with a **/\*\*** and ends with a **\*/**.

Ques no.2- What are the data types supported in java ?

Answer no.2-

* **PRIMITIVE DATA TYPES** – includes byte, short, int, long, float, double, Boolean and char.

• **INTEGERS**: This group includes **byte**, **short**, **int**, and **long**, which are for whole-valued signed numbers.  
• **FLOATING-POINT NUMBERS:** This group includes **float** and **double**, which represent numbers with fractional precision.  
• **CHARACTERS**: This group includes **char**, which represents symbols in a character set like letters and numbers.  
• **BOOLEAN :** This group includes **boolean**, which is a special type for representing true/false values.

|  |  |  |
| --- | --- | --- |
| Name | Width | Range |
| long | 64 | –9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |
| int | 32 | –2,147,483,648 to 2,147,483,647 |
| short | 16 | –32,768 to 32,767 |
| byte | 8 | –128 to 127 |

* **NON-PRIMITIVE DATA TYPES** – includes string, arrays, and classes.

Ques no.3 – What is the difference between char in C/C++ and char in java?

Answewr no.3 “Java char is an UTF-16 encoded Unicode code point while C uses ASCII encoding in most of the cases”.

* Java uses Unicode to represent characters.
* Java char is a 16-bit type.
* The range of a char is 0 to 65,536.
* There are no negative chars.
* But in C/C++ the size of **char** is 8 bit.

**Unicode defines a fully  
international character set that can represent all of the characters found in all human  
languages.**

Ques no. 4- What are the different types of operaters used in java?

Answer no.4- Operators are used to perform operations on variables and values.

**Types of operators:-**

* Arithmetic operator = +,-,\*,/,%,++,--.
* Assignment operators = = , +=.
* Comparison Operators = ==, >=, <=.
* Logical Operators = && , II, !.
* Bitwise Operators = & , I .
* Java provides a rich operator environment. Most of the operators can be divided intofour categories, They are,
* • Arithmetic Operators.
* • Bitwise Operators.
* • Relational Operators.
* • Logical Operators.
* Java also provides some special operators to handle some special situations. They are:-
* • Assignment Operators.
* • Ternary Operator / The ? Operator.
* **Arithmetic Operators**
* The Arithmetic Operators are used in mathematical expressions in the same way that they areused in algebra.

|  |  |
| --- | --- |
| **Operator** | **Result** |
| + | Addition (also unary plus). |
| - | Subtraction (also unary minus). |
| \* | Multiplication. |
| / | Division. |
| % | Modulus. |
| ++ | Increment. |
| += | Addition assignment. |
| -= | Subtraction assignment. |

|  |  |
| --- | --- |
| \*= | Multiplication assignment. |
| /= | Division assignment. |
| %= | Modulus assignment. |
| - - | Decrement. |

* **Bitwise Operators**
* Java defines several bitwise operators that can be applied to the integer types: long, int, short,char, and byte. These operators act upon the individual bits of their operands.

|  |  |
| --- | --- |
| **Operators** | **Result** |
| ~ | Bitwise unary NOT. |
| & | Bitwise AND. |
| | | Bitwise OR. |
| ^ | Bitwise exclusive OR. |
| >> | Shift right. |
| >>> | Shift right zero fill. |
| << | Shift left. |
| &= | Bitwise AND assignment. |
| |= | Bitwise OR assignment. |
| ^= | Bitwise exclusive OR assignment. |
| >>= | Shift right assignment. |
| >>>= | Shift right zero fill assignment. |
| <<= | Shift left assignment. |

* **Relational Operators**
* Relational operators determine the relationship that one operand has to the other. Specifically,they determine equality and ordering.

|  |  |
| --- | --- |
| **Operators** | **Result** |
| = = | Equal to. |
| ! = | Not Equal to. |
| > | Greater than. |
| < | Lesser than. |
| >= | Greater than or equal to. |
| <= | Lesser than or equal to |

* **Logical Operators**
* The Boolean logical operators shown here operate only on boolean operands. Most of thebinary logical operators combine two boolean values to form a resultant boolean value.

|  |  |
| --- | --- |
| **Operator** | **Result** |
| & | Logical AND. |
| | | Logical OR. |
| ^ | Logical XOR. |
| || | Short-circuit OR. |
| && | Short-circuit AND. |
| ! | Logical unary NOT |
| &= | AND assignment. |
| |= | OR assignment. |
| ^= | XOR assignment. |

* **Assignment Operator**
* The assignment operator is the single equal to symbol (=). The syntax of the assignmentoperator is,
* Syntax,variable = expression;
* Exampea=b+c;
* **Ternary Operator / The ? Operator**
* Java includes a special ternary operator that can be replaced certain types of if then-elsestatements. This operator is ?.
* Syntax,
* Example,
* expression1?expression:expression3
* int a = 5, b = 10;
* int greaterNo = a>b ? a : b

Ques No.5- Develop an interest interface which contains simple interest and compound interest method and static final field of rate 25%.

Answer no.5-

public class InterestCalculator{

private static final double RATE = 0.25; //25% interest rate

public static double calculateSimpleInterest(double principle, double timePeriod)

{

Double simpleInterest = (principle\*RATE\*timePeriod)/100;

return simpleInterest;

}

public static double calculateCompoundinterest(double principle, double timePeriod)

{

Double compundinterest = principle\*Math.pow(1+(RATE/100, timePeriod)-principle;

return compoundinterest;

}

Public static void main(String args[]){

double principle = 1000;

double timePeriod = 2;

double simpleInterest = calculateSimpleInterest(principle, timePeriod);

System.out.println(“Simple interest for principle”+principle+”and time period”+timePeriod+”years is”+simpleInterest);

Double compoundinterest = calculatecompoundinterest(principle,timePeriod);

System.out.println(“Compound interest for principle”+principle+”and time period”+timePeriod+”years is”+compoundinterest);

}

}

------------END OF ASSIGNMENT-------------