**Java class 01 revision for interview date; 24-08-2024**

Most popular languages

1. Java script -------🡪 why this is mostly used??

This is number one because it can be used on every browser. Fire fox, safari, chrome, internet explorer, opera, and edge.

After java script then come java language.

1. Java
2. Python
3. C++
4. C
5. C# sharp
6. Ruby

-------------------------------------------------------------------------------------------------------------------------------------------------------------

**What is programming language? Why we need programming language??**

How we talk with computers? we talk with them with programming languages, because computer does not understand human languages, they understand 0,1 binary language. Programming languages converts the human instruction to machine languages.

* Java is a high level computer programming language.
* Java is open source language, free
* Java is free to access and can run on all platforms.
* Java is a case-sensitive language.
* Java is an independent programming language that follows the logic of “Write once, Run anywhere” i.e. the compiled code can run on all platforms which supports java.

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**What is java???**

* Java is a high level programming language which is used to communicate with computer.

**Why do you prefer java???**

**I am an automation tester, to develop automation frameworks for websites , java language is mostly used. That’s why I prefer java.**

**Java**: Widely used in conjunction with frameworks like Selenium, TestNG, JUnit, and Cucumber. It's popular for web application testing and automation.

**Python**: Known for its simplicity, Python is often used with frameworks like PyTest, Robot Framework, and Behave. It's favored for both web and API testing.

**JavaScript**: Often used in front-end testing with frameworks like Jasmine, Mocha, and Jest, especially in Node.js environments. It's also used with Selenium for web automation testing.

**C#**: Used with testing frameworks like NUnit, MSTest, and SpecFlow, especially in .NET environments.

**Ruby**: Often paired with Cucumber and RSpec, Ruby is popular in BDD frameworks.

**PHP**: Used with PHPUnit for testing PHP applications.

-----------------------------------------------------------------------------------------------------------------------------------------------------------

**What is difference in java and other programming languages???**

When discussing the differences between Java and other programming languages, it's important to focus on the key aspects that set Java apart. Here's a comparison that you can use during an interview:

**1. Platform Independence:**

* **Java:** Java’s "Write Once, Run Anywhere" capability is a significant advantage. Java applications are compiled into bytecode, which can run on any device equipped with a Java Virtual Machine (JVM), making it platform-independent.
* **Other Languages:** Many languages, like C++ and C, are platform-dependent, requiring recompilation for different operating systems.

**2. Object-Oriented Programming (OOP):**

**Java:** Java is a pure object-oriented language where almost everything is an object. This OOP approach facilitates modular, reusable, and scalable code.

**Other Languages:** While many languages like C++, Python, and C# also support OOP, Java's approach is more consistent, with features like encapsulation, inheritance, and polymorphism being deeply integrated into the language’s design.

**3. Security:**

* **Java:** Java is designed with security in mind. It provides a secure runtime environment with built-in security features like the Java Security Manager, which can restrict what code can do, and bytecode verification.
* **Other Languages:** While many modern languages have strong security features, Java's security model is particularly robust and has been a focus since its inception, making it a preferred choice for applications where security is critical.

**Why there are so many languages????** Each language has its own purpose and functionality

| **Language** | **Purpose** | **Used For** | **Year Created** |
| --- | --- | --- | --- |
| C | System Programming | Making operating systems and embedded devices | 1972 |
| C++ | Application Development | Creating games and real-time systems | 1985 |
| Java | General Use | Building apps that work on many devices, Android apps | 1995 |
| Python | General Use | Data analysis, AI, websites, and automation | 1991 |
| JavaScript | Web Development | Making websites interactive and full-stack apps | 1995 |
| C# | Windows and Apps | Creating .NET apps and games | 2000 |
| Ruby | Web Development | Building web apps with Ruby on Rails | 1995 |
| PHP | Web Development | Writing scripts for websites | 1995 |

----------------------------------------------------------------------------------------------------------------------------------------------------------------

**What is java and when it was discovered???**

1. Open source free language
2. Case sensitive
3. High level language
4. Can run on all platforms

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**What are IDEs?**

IDE are note-pad or platform where developers and programmers write their code.

There are lot of ides but we will use only two

Eclipse – made by eclipse Corporation in 2001

Intelli j --- mad by jet brains in 2001.

**An integrated development environment (IDE) is a software application that provides facilities to computer programmers for software development.**

An IDE normally consists of a source code editor, compiler and a debugger.

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Compiler; is a tool which converts source code into bytecode.**

**JVM;** is a tool which is used to convert bytecode into machine read able and from machine code to bytecode.

**Debugger;** A debugger is a tool used to test and debug programs by allowing you to run code in a controlled environment.

**What is bytecode;** is intermediate code can run on any platform

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

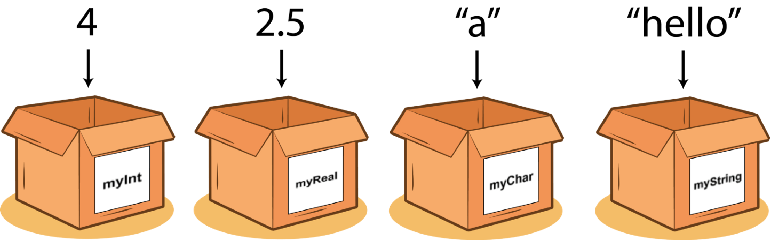
**Class 02**

**What is a data type??**

Data type specifies the **size** and type/kind of **values** that can be stored/hold in a variable.

We need different data types to hold different types of information.

Data structure means memory organized in computer.



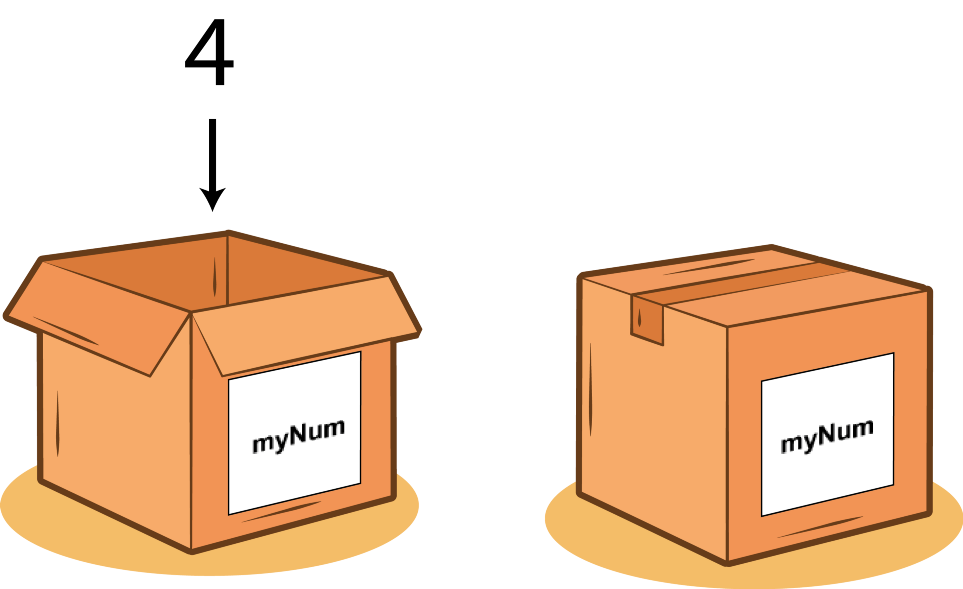
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**There are two data types??**

1. Primitive
2. Non primitive

--------------------------------------------------------------------------------------------------------------------------------------------------

**What is a variable???**

Variable is a name of memory location/container where we store data.**We store data inside a container. This container we call a VARIABLE**

**What does primitive mean?? What are primitive data types???**

It means old one, first one that java was introduced.

**Primitive data types is a data structure which is used to store only one type of data in stack area.**

Data types are not objects and are directly stored in memory with a fixed size. There are 8 primitive data types in java.

* **byte**: 127
* **short**: 32,767
* **int**: 2,147,483,647
* **long**: 9,223,372,036,854,775,807
* **float**: Approximately 3.4028235E38
* **double**: Approximately 1.7976931348623157E308
* **char**: 65,535
* **Boolean**: true or false

**------------------------------------------------------------------------------------------------------------------------------------------------------------**

**Non primitive data??**

Non-primitive data structure is a type of data structure that can store the data of more than one type in heap area.

Non-primitive data types are stored in the **heap** memory. The variable itself (the reference) is stored on the **stack**, but it points to the actual data stored on the heap. This is why they are called "reference types."

Examples of non-primitive data structure are Array, Linked list.

**Non-primitive data types can store data of more than one type. For example, a class can have multiple fields of different types (e.g., an int, a String, etc.). string can store combination “abc123.456A”**

**-----------------------------------------------------------------------------------------------------------------------------------------**

**Type casting??**

Assigning a value from one data type to a variable of another data type is called Type Casting.

**2 two types of casting in Java:**

1. Widening Casting (Implicit/ Automatic) - converting lower data type into higher data type is called widening

**int intValue = 100;**

**double doubleValue = intValue;**

1. Narrowing Casting(Explicit/ Manual) - converting higher data type into lower data type is called narrowing

double doubleValue = 9.78;

int intValue = (int) doubleValue;

----------------------------------------------------------------------------------------------------------------------------------------

**Why we do need type casting???**

* Reusing Existing Code
* Merging Code from Different Sources

the answer is that when someone write a program in double data type and after 4 year later you join the same company and you write your logic in integer data type then you have to use the code written 4 years earlier you have not too many time to write that code again in integer, you can convert his code by type casting and you can use easily.

Also code is not written by a single person in a company. Every one writes code according to his own logic and at the end all the code is merged together to get an application. During merging the code type casting helps.

----------------------------------------------------------------------------------------------------------------------------------------------

---------------------------------------------------------------------------------------------------------------------------------------------------------

**What is assignment operator????**

In Java, an assignment operator is used to assign a value to a variable. The most common assignment operator is the single equal sign (=). For example:

int x = 10;

In this case, the value 10 is assigned to the variable x.

--------------------------------------------------------------------------------------------------------------------------

**What are arithmetic operators??**

Arithmetic operators in Java are used to perform basic mathematical operations on numeric values.

There are five arithmetic operators in Java:

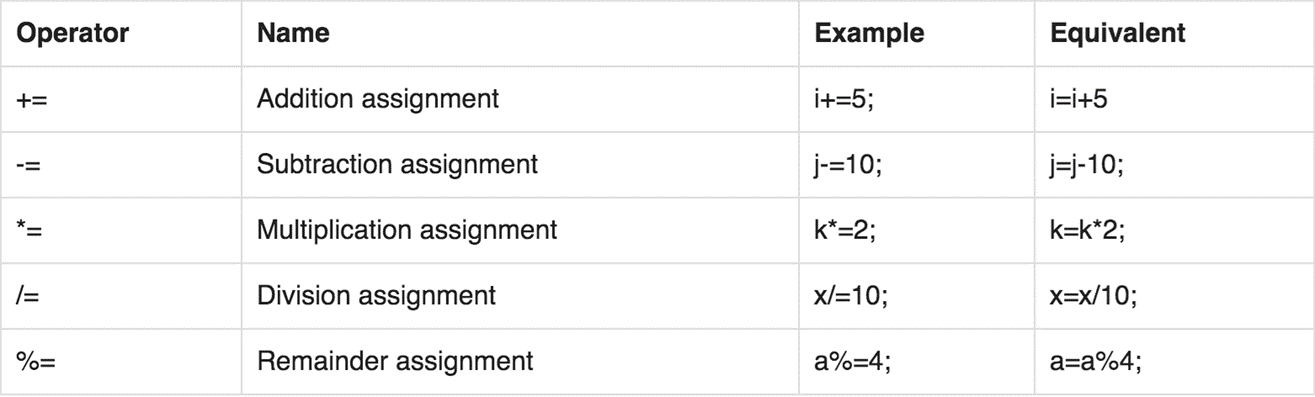
* **Addition (+)**: Adds two operands.
* **Subtraction (-)**: Subtracts the second operand from the first.
* **Multiplication (\*)**: Multiplies two operands.
* **Division (/)**: Divides the first operand by the second.
* **Modulus (%)**: Returns the remainder of the division of the first operand by the second.

In programming, an operand is a value or variable upon which an operator performs an operation. For example, in the expression 5 + 3, 5 and 3 are operands, and + is the operator

-------------------------------------------------------------------------------------------------------------------------------------------------

**What are compound operator/short hand operator??? --used only with primitive data except Boolean.**

**Compound Assignment (Shorthand) Operators** in Java are operators that combine an arithmetic operation with an assignment operation (=).



-----------------------------------------------------------------------------------------------------------------------------------------------

**What is increment operator?? --used only with primitive data except Boolean. ( its mean exponent of power \*\* and // and %% are not allowed as increment because multiply with 1 and divide with 1 no change in value. 12\*1=12 , and %% is valid in python. )**

In Java, increment and decrement operators are unary operators used to increase or decrease the value of a variable by one.

**1. Increment Operator (++)**

The increment operator increases the value of a variable by one.

* **Pre-increment (++variable)**
  + The value of the variable is incremented first, and then the incremented value is used in the expression.

int y = 5;

y = ++5; // y is incremented to 6, then y is assigned the value 6

* + After execution: y = 6
* **Post-increment (variable++)**
  + The current value of the variable is used in the expression, and then the variable is incremented.

int y = 5;

y = y++; // y is assigned the value 5, then x is incremented to 6

* + After execution: y = 6

**2. Decrement Operator (--)**

The decrement operator decreases the value of a variable by one.

* **Pre-decrement (--variable)**
  + The value of the variable is decremented first, and then the decremented value is used in the expression.
  + int y = 5;

y = --y; // y is decremented to 4, then y is assigned the value 4

* + After execution: y = 4
* **Post-decrement (variable--)**
  + The current value of the variable is used in the expression, and then the variable is decremented.

int x = 5;

x = x--; // x is assigned the value 5, then x is decremented to 4

* + After execution: x = 4

-----------------------------------------------------------------------------------------------------------------------------------------------------------

**What is relational operator and equality operator in java???? (give output boolean)**

Relational and equality operators are used to compare values in Java.

**Relational Operators**

1. **Greater than (a>b)**
   * **Description:** Checks if the left value is greater than the right value.
2. **Less than (a<b)**
   * **Description:** Checks if the left value is less than the right value.
3. **Greater than or equal to (a>=b)**
   * **Description:** Checks if the left value is greater than or equal to the right value.
4. **Less than or equal to (a<=b)**
   * **Description:** Checks if the left value is less than or equal to the right value.

**---------------------------------------------------------------------------------------------------------**

**Equality Operators**

1. **Equal to (a==b)**
   * **Description:** Checks if two values are equal.
2. **Not equal to (a!=b)**
   * **Description:** Checks if two values are not equal.

---------------------------------------------------------------------------------------------------------------------------------------------------------

**Math class??? Math is a class in java.**

All methods in the Math class are static, meaning you can call them directly on the Math class without creating an instance.

double power = Math.pow(2, 3); // 2 raised to the power of 3 = 8.0

double sqrt = Math.sqrt(16); // Square root of 16 = 4.0

double cbrt = Math.cbrt(27); // Cube root of 27 = 3.0

double num1 = 4.3; double result1 = Math.ceil(num1); // Rounds up to 5.0

double num1 = 4.7; double result1 = Math.floor(num1); // Rounds down to 4.0

double num = 4.7; System.out.println(Math.round(num)); // Output: 5 System.out.println(Math.floor(num)); // Output: 4.0 System.out.println(Math.ceil(num)); // Output: 5.0

-------------------------------------------------------------------------------------------------------------------------------------------------------------

**What is String format??**

The String.format() method in Java belongs to the String class. The String class is part of the java.lang package, which is automatically imported into every Java program.

double value = 12.34567;

String result = String.format("%,.2f", value); // Formats to two decimal places

System.out.println(result); // Output: 12.35

**%f - Floating-Point Numbers**

* **Description:** Formats floating-point numbers (both float and double).

**%,d** for integer, byte, short, long. Only use %d digit which mean it will add comma after 1k value only.

**%40s** for string format, here 40 is width, your text will print in console in center.

------------------------------------------------------------------------------------------------------------------------------------------------------

**What are flow control statement in java????**

Flow control in Java are used to control the flow of a program, how and when different parts of the code will be executed.

Flow control statement allows the program to make decisions (if, switch), repeat actions (for, while, do-while), and jump to different parts of the code (break, continue, return method).

-------------------------------------------------------------------------------------

Flow control statements in Java are generally **categorized into three main types**:

1. **Conditional Statements**: Used to make decisions based on conditions.
   * if
   * if-else
   * if-else if-else
   * switch
2. **Loop Statements**: Used to repeat actions as long as a condition is met.
   * for
   * while
   * do-while
   * Enhanced for (for-each) loop
3. **Jump Statements**: Used to alter the flow of control by breaking out of loops or switching execution.
   * Break – jump out from the loop
   * continue – skip one iteration
   * return -Exits from the current method and returns control to the method that called it

----------------------------------------------------------------------------------------------------------------------------------------------------

**When to use if and if else condition???**

1. If condition ------------ > is used when there is only one condition
2. Else -------- > is optional and also this is second condition if first condition is fails then what do next???

**When to use nested if condition??? When one condition is dependent on other condition.**

1. Nested if condition ------------- > this is particularly useful when you have a set of conditions that depend on each other. if one condition pass then you will check other condition inside if statement

You can have many nested if else if condition inside nested if condition.

**------------------------------------------------------------------------------------------------------------------------------------------------------------**

**Class 05 java date;08-12-2023**

**What is scanner class in java???**

**Scanner class is used to read inputs from keyboard.**

The Scanner class is used to get user input, and it is found in the java.util package.

The Scanner class is used for reading in primitive data types like int, double, float, etc., and objects of type String.

* To read strings, we use nextLine() or next()
* To read number values, we use nextInt()
* To read decimal values, we use nextDouble()
* To read a single character, we use next().charAt(0)

**Why logical operators are used?????????????**

When we want to test 2 or more than 2 conditions in one statement. if (true && true)

**What are logical operators?????????**

Logical operators are symbols or words used in programming to perform logical operations.

**There are three logical operators;**

1. **And operator**
2. **Or operator**
3. **Not operator**

**Boolean has two variations true and false. Logical operator depends on Boolean values.**

**And operator**

**True && true = true**

**True && false = false**

**false && true = false**

**false && false = false**

**The symbol & is called an ampersand**

**-----------------------------------------------------------------------------------------------------------------------------**

**or || operator ----** **The symbol || is called a double pipe or a double vertical bar**

**false || false = false**

**True || false = true**

**false || true = true**

**true || true = false**

**Not ! symbol--------! is called an exclamation mark or a bang**

**True ! = false**

**False !=true**

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Class 06**

**When use switch condition?????**

A **switch statement** in Java is a control flow’s conditional statement that allows you to execute one block of code among many based on the value of a given expression. It's an alternative to using multiple if-else statements

**switch (expression)** {

**case value1:** // Code block for value1

**break;** // Optional, exits the switch statement

**How it Works:**

1. **Expression:** The expression inside the switch statement is evaluated once.
2. **Cases:** The result of the expression is compared with the values of each case. If a match is found, the corresponding code block is executed.
3. **break Statement:** The break statement is used to exit the switch block once a matching case is executed. Without break, the execution will continue to the next case (known as "fall-through").
4. **default Case:** If no matching case is found, the default case is executed. It acts as a catch-all.

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Limitations of switch statement**

* The switch can only check for equality. This means that the other relational operators such as greater than are rendered unusable in a case.

case k>=20: // not allowed

* Logical operators cannot be used with switch statement.
* Primitive types: Boolean, float, double, long cannot be used as an argument in switch ()

Switch case is faster and more readable. it is used when you have to check value of a variable against multiple constant values.

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Why we use loops??**

**Scenario**

You have a script written, if you run your program without the loop it will only run once. What if you want to run that script 20 times? Would you like to wait for the program to finish so you can click the run button again IMAGINE you have to repeat this step for 20 times. **IT WILL BE A WASTE OF TIME**

------------------------------------------------------------------------------------------------------------------------------------------

**What is a loop???**

In programming, a **loop** is a control flow statement that is used to execute a block of code repeatedly based on a condition.

**What is a while loop???**

The while loop in Java is a control flow statement that is used when you don't know the exact number of iterations and want to keep looping until a specific condition is met.

While is keyword in java , which mean repeat the block of code.

**Do – while loop??????**

In Java, the **do-while** loop is a flow control statement is used when you want to execute a block of code at least once, and then the condition is checked.

This ensures that the block of code is executed at least once, even if the condition is initially false.

Do {

Block of code

} while (condition)

Lets create a secrete number from 1 to 100; with scanner class is best example of do while loop,

------------------------------------------------------------------------------------------------------------------------------------------

**When use for loop???????**

The for loop in Java is a control flow statement that is used when you already know the exact number of iterations in advance and want to keep looping until a specific condition is met.

In one line of code there are three condition. You can say everything in single line. But in while loop the scenario is different

1. initialization

2. condition

3. increment/decrement

Say 4 times good morning.

For (int a=0; a>4; a++) {}

**What is break keyword??**

The Java ***break*** is used to break loop or switch statement. It breaks the **current flow** of the program at specified condition. In case of inner loop, it breaks only inner loop.

**What is continue keyword????**

The Java *continue* keyword can be used in any of the loop control structures. It causes the loop to immediately jump to the next iteration of the loop.

--------------------------------------------------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------------------------------

**What is an array??????**

In Java, an array is a data structure/non primitive data type that is used to store multiple values of the same type in a single variable.

It is a collection of similar data types.

It is fixed in size that means you can't increase the size of array at run time.

It stores the value on the basis of index value. The first element of an array starts with zero

-----------------------------

1) At the time of array creation we must be specify the size of array otherwise get an compile time error. For Example   
int[] a = new int[]; → Invalid.  
int[] a = new int[5]; → Valid

2) If we specify array size as negative int value, then we will get run-time error, NegativeArraySizeException.

3) To specify array size the allowed data types are byte, short, int, char. If we use other data type then we will get an compile time error.

4) The maximum allowed size of array in java is 2147483647

(It is maximum value of int data type)

--------------------------------------------------------------------------------------------------------

Arrays in Java are objects of special classes that are created automatically by the Java Virtual Machine (JVM).

---------------------------------

**Key Features of Arrays in Java:**

1. **Fixed Size**: Once an array is created, its size cannot be changed. You must specify the size of the array when you create it.
2. **Homogeneous Elements**: All elements in an array must be of the same type, such as integers, strings, or objects of a specific class.
3. **Indexed base**: Elements in an array are accessed using their index, allowing for quick retrieval and modification of values.
4. **Memory Layout**: Arrays are stored in contiguous memory locations, making access to elements fast due to predictable addressing.

-------------------------------------------------------------------------------------------------------------------------------------------------------

Default values of primitive data type??

**Primitive Data Types**:

* **byte**: 0
* **short**: 0
* **int**: 0
* **long**: 0L
* **float**: 0.0f
* **double**: 0.0d
* **char**: '\u0000' (null character, equivalent to 0)
* **boolean**: false

**Reference Types**:

* **String and all other objects**: null

-----------------------------------------------------------------------------------------

**Fixed Size**: If an array's length is 20, memory is allocated for 20 elements, not just for the initialized ones.

**Unused Elements**: If you only use 10 elements, the remaining 10 elements still occupy memory because the array reserves space for all 20 elements.

-----------------------------------------------------------------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Why we use nested loops???????**

This is often necessary when dealing with two-dimensional arrays, matrices, or nested data structures.

---------------------------------------------------------------------------------------------------------------------------------------------------------------

============================================================================================================================================================================================

**What does System.out.println() do in Java?**

**Answer:** System.out.println() is used to print a message to the console or standard output. It prints the argument passed to it and then moves the cursor to the next line. It is commonly used for debugging and displaying output in Java programs.

**Explain the components of System.out.println ().???**

**Answer:**

* **System:** A class from the java.lang package that provides access to system-level resources.
* **out:** A static variable within the System class, representing the standard output stream (an instance of PrintStream).
* **Println ():** A method of the Print Stream class that prints the argument passed to it and adds a newline at the end.

**How does System.out.println() handle different data types?**

Answer: System.out.println() is an overloaded method in the Print Stream class, meaning it has different versions that can handle various data types, such as int, char, boolean, float, double, String, and objects. If you pass an object, it calls the object's toString() method to get a string representation of the object.

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------