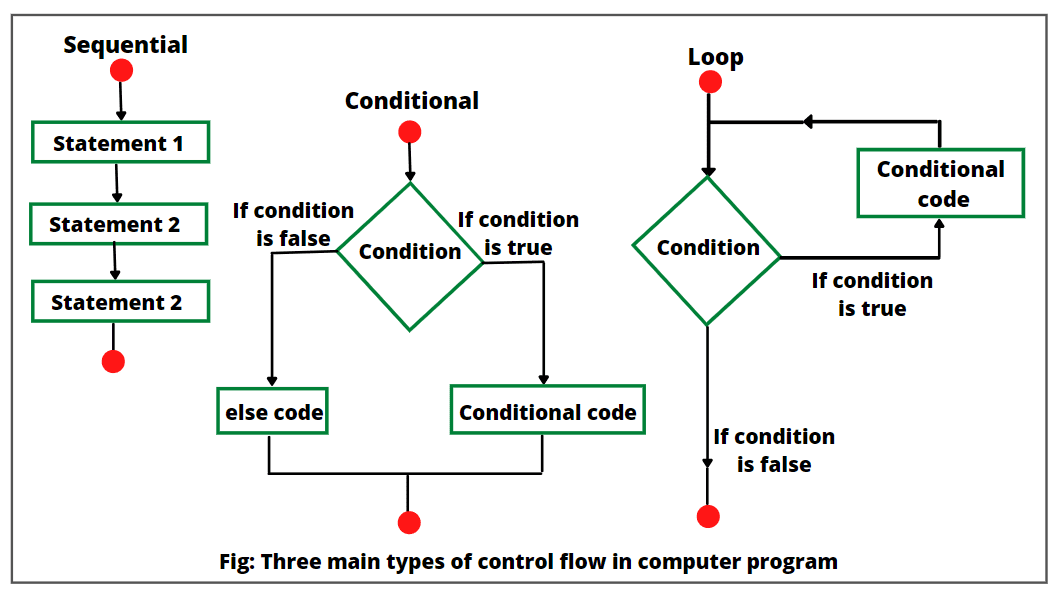
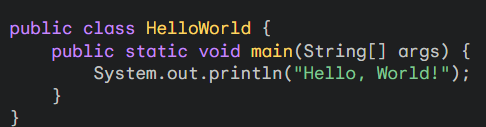
**Topic:** Java History and Basic Code Flow  
  
**Introduction**

Java, a high-level, object-oriented programming language, has revolutionized the software development landscape.   
Its versatility, platform independence, and robust ecosystem have made it a preferred choice for developers worldwide.  
  
**A Brief History of Java**

**Birth:** Created in 1995 by Sun Microsystems (now owned by Oracle),  
Java was initially designed for interactive television but was later adapted for the internet.  
**Platform Independence:** The "Write Once, Run Anywhere" (WORA) principle of Java allows compiled Java code to run on any system with a Java Virtual Machine (JVM) installed.  
**Object-Oriented Paradigm**: Java embraces the object-oriented programming (OOP) paradigm, which models real-world entities as objects, making code more organized and reusable.

**Basic Code Flow**Java code is written in .java files and compiled into bytecode, which can be executed on any platform using the Java Virtual Machine (JVM). A typical Java program consists of the following components:  
**Class Definition:** A class is a blueprint for creating objects. It contains data (attributes) and methods (behaviors).  
**Main Method:** The main method is the entry point of a Java program. Execution begins from here.  
**Variables:** Variables are used to store data values. They have a data type (e.g., int, double, String) and a name.  
**Operators:** Operators are symbols used to perform operations on variables (e.g., arithmetic, comparison, logical).  
**Control Flow Statements:** These statements control the order in which instructions are executed. Common examples include:  
**Conditional statements:** if, else if, else  
**Loops:** for, while, do-while  




**public class HelloWorld:** Defines a public class named HelloWorld.  
**public static void main(String[] args):** The main method, which is the entry point of any Java application.  
**System.out.println("Hello, World!");** Prints “Hello, World!” to the console.  
  
  
 **Interview Questions**

. **What is java and features?**

\* A programming language created by sun micro system and it was spelled as Oak before   
 public release  
\* All the instructions in one place are called as program and it is MLL  
 \* **Features:** simple OOP Portable Platform Independent Secured Robust Performance Threads  
 **\* Why it is Platform independent?**

\* Platform: - combination of both processor + operating system  
\* java compiler converts into byte code, JVM is all into binary  
 \* WORA write once run anywhere  
 \* we can use the byte code and can use in any system   
  
\* Interpreter: convert the code into MLL line by line

\* Compiler: it takes HLL input and convert to MLL. A software which converts HLL to MLL

Source: whatever we have coded and saved like (sysout, class), once it is saved it compiled to byte is known as class   
   
\*Class: which consists of java compiled code or bytes code  
 Design or prototype from which objects are created  
 class has variables and methods  
 methods have brackets ()

Object: Instance of class in JDK we have JRE (lib) -> JVM  
  
**Detailed explanation about class and object**

class is person (BLUEPRINT)

Compiler: manager: will talk i want table which suit my gaming chair so this manager change the   
 details into technical terms  
 JVM(Carpenter): Object (Table) Technical person width heigh material which type of table

This blue print gets converted into bytecode and this bytecode goes into JVM it will give object

1. i went to the carpenter and ask him i want table

2. he said which type of table u want (height, W, Material)

3. I didnt get him so i ask manager i want table which suits my gaming chair

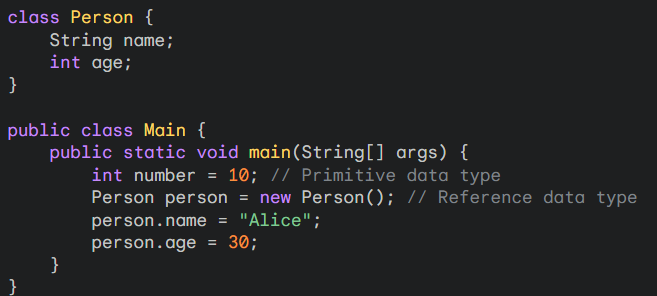
4. The manager (Compiler) changed my details into technical terms  
  
Java Data Types

|  |  |  |
| --- | --- | --- |
| Data Type | Size (bytes) | Range |
| byte | 1 | -128 to 127 |
| short | 2 | -32,768 to 32,767 |
| int | 4 | -2,147,483,648 to 2,147,483,647 |
| long | 8 | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |
| float | 4 | Approximately ±3.4E-38 to ±3.4E+38 |
| double | 8 | Approximately ±1.7E-308 to ±1.7E+308 |
| char | 2 | Unicode character (0 to 65,535) |
| boolean | 1 | true or false |

**Reference Data Types**

Reference data types refer to objects in memory. They don't store the actual values but rather a reference to the location where the object is stored. Examples of reference data types include:

* **Classes:** Custom data types defined by the programmer.
* **Arrays:** Ordered collections of elements of the same type.
* **Interfaces:** Contracts that define a set of methods that implementing classes must provide.



In this example, number is a primitive data type, while person is a reference data type. person refers to an object of the **Person** class.  
  
**Java Syntax**

- **Variables**: int x = 5;

- **Data types**: byte, short, int, long, float, double, boolean

- **Operators:** x = 5 + 3;

- **Control flow**: if (x > 10) { System.out.println("x is greater than 10"); }

**Interview Questions**

**Basic Understanding**

1. What are the two main categories of data types in Java?  
Primitive data types: Built-in to the language, representing basic values (e.g., int, double, char).  
Reference data types: Refer to objects in memory (e.g., String, Array, Class).

2. Explain the difference between primitive and reference data types.  
Primitive data types: Store values directly, have fixed sizes, and are passed by value.  
Reference data types: Store references to objects, have variable sizes, and are passed by reference.  
  
**Specific Data Types**

3. What is the default value for a boolean variable?  
false.

4. When should you use a float over a double?  
Use float when you need to conserve memory or when you require lower precision. double is generally more precise and should be used for most calculations.

5. What is the difference between a char and a String?  
char: Represents a single character (e.g., 'A').  
String: Represents a sequence of characters (e.g., "Hello").

**Data Type Conversions**

6. How do you convert a double to an int?  
Use a type cast: int x = (int) myDouble; This will truncate the decimal part.

7. What is the difference between implicit and explicit type conversion?  
Implicit: Automatic conversion by the compiler (e.g., from int to double).  
Explicit: Manual conversion using a type cast (e.g., from double to int).

**Memory Usage**

8. Which data type occupies the most memory?  
double.

9. How can you determine the size of a data type in Java?  
Use the sizeof() method from the sun.misc package. However, this method is not guaranteed to be available in all Java environments.

**Advanced Topics**

10. What is autoboxing and unboxing?  
Autoboxing: Automatic conversion between primitive types and their corresponding wrapper classes (e.g., int to Integer).  
Unboxing: Automatic conversion between wrapper classes and their corresponding primitive types (e.g., Integer to int).

11. What are generic types in Java?  
Generic types allow you to create classes and methods that can work with different data types.

12. How do you declare a variable of a generic type?  
List<String> myList = new ArrayList<>();.