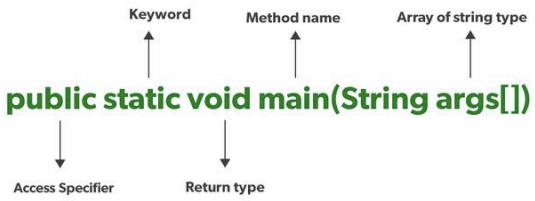
# JAVA : ASSIGNMENT 1

1) Explain why the main function is declared as “public static void main(String args[])”?
2) Explain “Write once and run anywhere” nature of Java.
3) The Java Archive (JAR) file format enables you to bundle multiple files into single archive file. What are the benefits of making JAR file?
4) Java supports automatic type conversion for variables. Explain widening conversion and narrowing conversion?
5) Write a note on JVM as emulator.
6) Explain package visibility control mechanism in Java
7) There is no operator overloading in Java? Why Not?
8) Explain Sandbox Model
9) What is type code in Java
10) Why Java is called strongly typed language?
11) Explain the advantages of Unicode over ASCII.
12) Explain the JVM architecture.
13) Java is platform independent . Justify this statement.
14) Explain characterstics of Java
15) Expalin Garbage collection in Java
16) What is meant by ByteCode inJava?
17) Explain about creation of classes and creation of objects inJava with example.
18) Write short note on:-
    a. Security promises of JVM
    b. Wrapper classes
19) Differentiate between JAVA and C++ Object Oriented Language
20) Fille in the blankes
    a. Converting a primitive data type into its corresponding wrapper class object instance is called_____
    b. Methods that have same name , but different parameter list and different definition is known as _____
    c. In java , gc() method for garbage collection is available in _____package

Q1 Explain why the main function is declared as “public static void main(String args[])”?

- In Java programs, the point from where the program starts its execution or simply the entry point of Java programs is the **main()** method
- It is one of the most important methods of Java and having a proper understanding of it is very important.
- The Java compiler or JVM looks for the main method when it starts executing a Java program.




## 1. Public

It is an *Access modifier*, which specifies from where and who can access the method. Making the *main()* method public makes it globally available. It is made public so that JVM can invoke it from outside the class as it is not present in the current class.

## 2. Static

When java runtime starts, there is no object of the class present. That’s why the main method has to be static so that JVM can load the class into memory and call the main method. If the main method won’t be static, JVM would not be able to call it because there is no object of the class is present.

### 3. Void

It is a keyword and is used to specify that a method doesn't return anything. As the *main()* method doesn't return anything, its return type is *void*. As soon as the *main()* method terminates, the java program terminates too. Hence, it doesn't make any sense to return from the *main()* method as JVM can't do anything with the return value of it.

### 4. Main

It is the name of the Java main method. It is the identifier that the JVM looks for as the starting point of the java program. It's not a keyword.

### 5. String Args

It stores Java command-line arguments and is an array of type ***java.lang.String*** class. Here, the name of the String array is *args* but it is not fixed and the user can use any name in place of it.

## Q2 Explain "Write once and run anywhere" nature of Java.

In traditional programming languages like C, C++ when programs were compiled, they used to be converted into the code understood by the particular underlying hardware, so If we try to run the same code at another machine with different hardware, which understands different code will cause an error, so you have to re-compile the code to be understood by the new hardware

.In Java, the program is not converted to code directly understood by Hardware, rather it is converted to bytecode(.class file), which is interpreted by JVM, so once compiled it generates bytecode file, which can be run anywhere (any machine) which has JVM( Java Virtual Machine) and hence it gets the nature of Write Once and Run Anywhere.

## Q3 The Java Archive (JAR) file format enables you to bundle multiple files into single archive file. What are the benefits of making JAR file?

A JAR (Java Archive) is a package file format typically used to aggregate many Java class files and associated metadata and resources (text, images, etc.) into one file to distribute application software or libraries on the Java platform.
A JAR file allows Java runtimes to efficiently deploy an entire application, including its classes and their associated resources, in a single request. JAR file elements may be compressed, shortening download times
Jar files can be easily imported in a project library and seamlessly used in multiple projects.

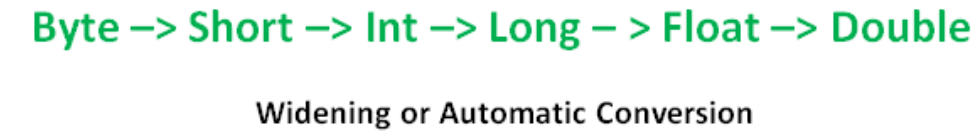
## Q4 Java supports automatic type conversion for variables. Explain widening conversion and narrowing conversion?

### Widening or Automatic Type Conversion

Widening conversion takes place when two data types are automatically converted. This happens when:

- The two data types are compatible.
- When we assign a value of a smaller data type to a bigger data type.

For Example, in java, the numeric data types are compatible with each other but no automatic conversion is supported from numeric type to char or boolean. Also, char and boolean are not compatible with each other.




Widening or Automatic Conversion

**Narrowing or Explicit Conversion**


Narrowing or Explicit Conversion

Converting a higher data type into a lower one is called **narrowing** type casting. It is also known as **explicit conversion** or **casting up**. It is done manually by the programmer. If we do not perform casting then the compiler reports a compile-time error.

## Q5 Write a note on JVM as emulator.

JVM is sometimes called an emulator **because it emulates hardware**, but in fact is software.

JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist. It is a specification that provides a runtime environment in which Java bytecode can be executed. It can also run those programs which are written in other languages and compiled to Java bytecode.

THERE ARE 4 TYPES OF ACCESS SPECIFIERS

- **default** - only accessible within the package
- **`private`** - only accessible in the class
- **`protected`** - accessible to classes in package and subclasses outside package
- **`public`** - always accessible

## Q6 Explain package visibility control mechanism in Java

When all classes are in the same package:

- A `private` variable is not visible from an object of its own class. But a default, `protected`, or `public` variable is visible from an object. This is true for `static` variables as well.
- The above is true even if trying to access the variable from a subclass.

When classes are in different packages:

- A `private` is accessible through a public method across packages.
- A default, `protected`, or `private` is not accessible across packages.

## Q7 There is no operator overloading in Java? Why Not?

**Programming error -**Java doesn't allow user defined operator overloading because if you allow programmer to do operator overloading they will come up with multiple meanings for same operator which will make the learning curve of any developer hard and things more confusing and messing.

**Makes code complex** – In case of operator overloading the compiler and interpreter (JVM) in Java need to put an extra

effort to know the actual functionality of the operator used in a statement.
**Method overloading** – The functionality of operator overloading can be achieved in Java using method overloading in Java in a simple, error free and clear manner.

## Q8 Explain Sandbox Model

- A sandbox typically provides a tightly controlled set of resources for guest programs to run in, such as limited space on disk and memory.
- In a Java programming language, the sandbox is the program area and it has some set of rules that programmers need to follow when creating Java code (like an applet) that is sent as part of a page.
- Since a Java applet is sent automatically as part of the page and can be executed as soon as it arrives, the applet can easily do harm, either accidentally or intentionally
- The sandbox restrictions provide strict limitations on which system resources the applet can request or access.The programmer must write code that “plays” only within the sandbox, such as children are allowed to play within the confined limits of a place.

## Q10 Why is Java a strongly typed language?

Java is considered strongly typed because it demands the declaration of every variable with a data type. Users cannot create a variable without the range of values it can hold. Once declared, the data type of the variable cannot be changed.

# Q11 Explain the advantages of Unicode over ASCII

Unicode and ASCII are the most popular character encoding standards that are currently being used all over the world. Unicode is the universal character encoding used to process, store and facilitate the interchange of text data in any language while ASCII is used for the representation of text such as symbols, letters, digits, etc. in computers.

- Global source and binary.
- Support for mixed-script computing environments.
- Improved cross-platform data interoperability through a common codeset.

**Size –**

1. Unicode represents far more characters than ASCII. ASCII uses a 7-bit range to encode just **128** distinct characters. Unicode on the other hand encodes **154** written scripts. And did I mention emoji? Those too.
2. So, we can say that, while Unicode supports a larger range of characters it also takes up a lot more space than ASCII.

• Simplified application development

Unicode is becoming the universal code page of the Web. Current Web standards require Unicode and rely on it.

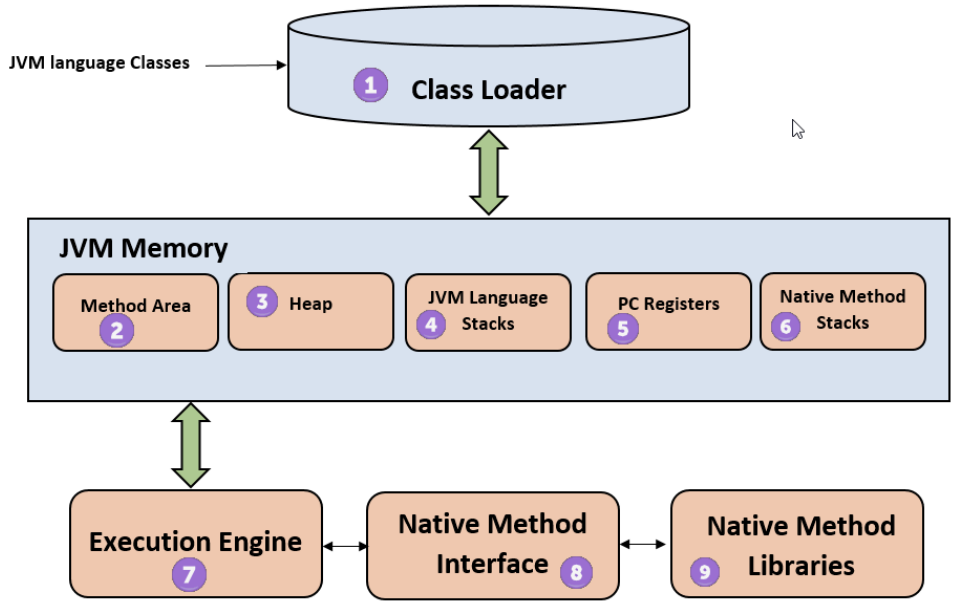
Applications using Unicode can support multiple languages in:

- Data
- User interface
- Reports

# Q12 Explain the JVM architecture.

JVM(Java Virtual Machine) acts as a run-time engine to run Java applications. JVM is the one that actually calls the **main** method present in a java code. JVM is a part of JRE(Java Runtime Environment).

When we compile a *.java* file, *.class* files(contains byte-code) with the same class names present in *.java* file are generated by the Java compiler. This *.class* file goes into various steps when we run it. These steps together describe the whole JVM.




## 1) ClassLoader

The class loader is a subsystem used for loading class files. It performs three major functions viz. Loading, Linking, and Initialization.

## 2) Method Area

JVM Method Area stores class structures like metadata, the constant runtime pool, and the code for methods.

**3) Heap**

All the Objects, their related instance variables, and arrays are stored in the heap. This memory is common and shared across multiple threads.

**4) JVM language Stacks**

Java language Stacks store local variables, and it's partial results. Each thread has its own JVM stack, created simultaneously as the thread is created. A new frame is created whenever a method is invoked, and it is deleted when method invocation process is complete.

**5) PC Registers**

PC register store the address of the Java virtual machine instruction which is currently executing. In Java, each thread has its separate PC register.

**6) Native Method Stacks**

Native method stacks hold the instruction of native code depends on the native library. It is written in another language instead of Java.

**7) Execution Engine**

It is a type of software used to test hardware, software, or complete systems. The test execution engine never carries any information about the tested product.

**8) Native Method interface**

The Native Method Interface is a programming framework. It allows Java code which is running in a JVM to call by libraries and native applications.

**9) Native Method Libraries**

Native Libraries is a collection of the Native Libraries(C, C++) which are needed by the Execution Engine.

**Q13 Java is platform independent . Justify this statement.**

- Whenever, a program is written in JAVA, the javac compiles it.
- The result of the JAVA compiler is the **.class file or the bytecode** and not the machine native code (unlike C compiler).
- The bytecode generated is a non-executable code and needs an interpreter to execute on a machine. This interpreter is the JVM and thus the Bytecode is executed by the JVM.
- And finally program runs to give the desired output.

**it is the magic of Bytecode that makes it platform independent**

The byte code generated by source code compilation would run in any operating system, but the JVM present in a machine differs for each operating system. And this is how java is considered a platform-independent programming language.
Java Virtual Machine acts as an interpreter and then executes the byte code generated by javac

# Q14 Explain characterstics of Java

**Simple :**

- Java is Easy to write and more readable and eye catching.
- Java has a concise, cohesive set of features that makes it easy to learn and use.
- Most of the concepts are drew from C++ thus making Java learning simpler.

**Secure :**

- Java program cannot harm other system thus making it secure.
- Java provides a secure means of creating Internet applications.
- Java provides secure way to access web applications.

**Portable :**

- Java programs can execute in any environment for which there is a Java run-time system.(JVM)
- Java programs can be run on any platform (Linux,Window,Mac)
- Java programs can be transferred over world wide web (e.g applets)

**Object-oriented :**

- Java programming is object-oriented programming language.
- Like C++ java provides most of the object oriented features.
- Java is pure OOP. Language. (while C++ is semi object oriented)

**Robust :**

- Java encourages error-free programming by being strictly typed and performing run-time checks.

**Multithreaded :**

- Java provides integrated support for multithreaded programming.

**Architecture-neutral :**

- Java is not tied to a specific machine or operating system architecture.
- Machine Independent i.e Java is independent of hardware .

**Interpreted :**

- Java supports cross-platform code through the use of Java bytecode.
- Bytecode can be interpreted on any platform by JVM.

**High performance :**

- Bytecodes are highly optimized.
- JVM can executed them much faster .

**Distributed :**

- Java was designed with the distributed environment.
- Java can be transmit,run over internet.

**Dynamic :**

- Java programs carry with them substantial amounts of run-time type information that is used to verify and resolve accesses to objects at run time.

## Q15 Expalin Garbage collection in Java

Garbage Collection is process of reclaiming the runtime unused memory automatically. In other words, it is a way to destroy the unused objects.

Garbage collection in Java is the process by which Java programs perform automatic memory management. Java programs compile to bytecode that can be run on a Java Virtual Machine, or JVM for short. When Java programs run on the JVM, objects are created on the heap, which is a portion of memory dedicated to the program. Eventually, some objects will no longer be needed. The garbage collector finds these unused objects and deletes them to free up memory.
In java, garbage means unreferenced objects.