```
// What is Platform Independence?

// Why is Java so Popular?

// Difference between C++ and Java?
// * Pointers
// * Structures and Unions
// * Memory Management vs Garbage Collection
// * Multiple Inheritance
// * Java if fully Object Oriented Language
```

JVM

Platform independent
Pure Object oriented easy to maintain

Pointers: reference to special value in store memory. The address is called a pointer.
Structures and Unions Class(structure and method) and Objects

C++ Dynamic allocate memory ;
Java automate and garbage collections

Multiple inheritance
Pure objective oriented programming.

Encapsulation
Abstraction
Inheritance
Polymorphism

```
// What are the advantages of OOPS?
// * Domain Mapping
// * Reuse
```

Reuse

Java is fully Passed by value or Pass by reference
String builder is not synchronized
Static and instance
An static variable is shared by all instance of the class.

	Class	Package	Subclass	World
public	y	y	y	y
protected	y	y	y	n
no modifier	y	y	n	n
private	y	n	n	n

y: accessible
n: not accessible

An abstract class, in contrast, provides more structure. It usually defines some default implementations and provides some tools useful for a full implementation.

`extends` is for *extending* a class.

`implements` is for *implementing* an interface