**What is Java?**

Java ဆိုတာက object-oriented, class-based, concurrent, secured and general-purpose computer-programming language တစ္ခုၿဖစ္တယ္။သူက က်ယ္က်ယ္ၿပန္႔ၿပန္႔သံုးေနတဲ့ robust technology တစ္ခုလည္းၿဖစ္တယ္ (robust technology ကိုေနာက္ပိုင္းၿပန္ရွင္းပါမယ္) ။

**Java ဆိုတာဘာလဲ**

Java ဆိုတာ programming language တစ္ခုၿဖစ္သလို platform တစ္ခုလည္း ၿဖစ္တယ္။

Platform ဆိုတာဘာလဲ: မည္သည့္ hardware သို႔မဟုတ္ software တစ္ခုခုေပၚမွာ program တစ္ခုကို run ႏိုင္တယ္ဆိုရင္ အဲ့ environment ကို platform လို႔ေခၚႏိုင္တယ္။

**Types of Java Application**

၄ မ်ိဳးရွိတယ္။

#### 1) **Standalone Application:** desktop application မ်ားနဲ႔ window application မ်ားၿဖစ္တယ္။ သူတို႔က traditional software မ်ားၿဖစ္ၿပီးေတာ့ သူတို႔ကို မည္သည့္ စက္တိုင္းမွာမဆို install လုပ္ႏိုင္ရမယ္။

2) **Web Application:** server side ေပၚမွာလည္း run မယ္ dynamic pages မ်ားလည္း ၿဖစ္ေနမယ္ဆိုရင္ အဲ့တာကို web application တစ္ခုလို႔ ေခၚႏိုင္တယ္။ Servlet, JSP, Struts, spring, Hibernate, JSF, etc. အဲ့ technologies ေတြကို အသံုးၿပဳလို႔ရတယ္။

3**) Enterprise Application:** distribution သေဘာ application ကို ဆိုလိုၿခင္းၿဖစ္တယ္ ဥပမာ banking application လိုဟာမ်ိဳးေပါ့။ သူမွာ ဘာအားသာခ်က္မ်ိဳးရွိေတာ့ security ေကာင္းတယ္ performace ေကာင္းတယ္ clustering ၿဖစ္တယ္။ Enterprise application ကို EJB technologies ကိို အမ်ားဆံုးသံုုးၾကတယ္။

4) **Mobile Application:** android ေပါ့။

**Java Platforms သို႔မဟုတ္ Java Editions**

1. **Java SE (Java Standard Edition):** သူ႕ထဲမွာ Java Programming APIs မ်ားစြာပါ၀င္တယ္ ဘာေတြလဲဆိုေတာ့ java.lang, java.io, java.net, java.util, java.sql, java.math etc ၿဖစ္ပါတယ္။အဲ့ APIs ထဲမွာဘာေတြပါလဲဆိုေတာ့ OOPs, String, Regex, Exception, Inner classes, Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection, etc တို႔ ပါ၀င္ပါတယ္။
2. **Java EE (Java Enterprise Edition):** web သို႔မဟုတ္ enterprise applications မ်ားကို develop လုပ္ဖို႔အတြက္သံုးပါတယ္။သူ႕ထဲမွာ Servlet, JSP, Web Services, EJB, JPA, etc တို႔ပါ၀င္ပါတယ္။
3. **Java ME (Java Micro Edition):** mobile Application အတြက္ေပါ့။
4. **Java FX:** light weight use interface API အတြက္ အသံုးၿပဳပါတယ္။

**Features of Java**

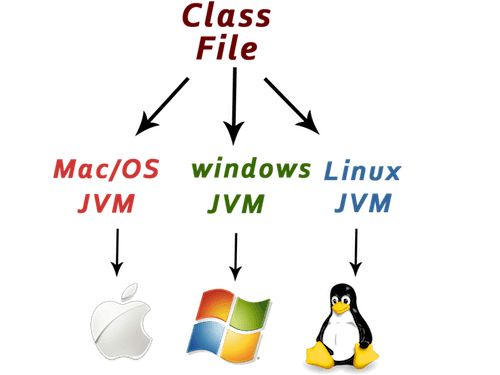
Java မွာ အေရးပါတဲ့ ၁၂ ခုရွိပါတယ္။



**Object-oriented:** java ဆိုတာက object-oriented programming language တစ္ခုၿဖစ္တယ္။ Java မွာ အရာရာတိုင္းမွာ object ၿဖစ္တယ္။ object-oriented ဆိုတာက ကၽြန္ေတာ္တို႔ရဲ႕ software ၾကီးဆိုတာက မတူညီတဲ့ objects မ်ားကို စုေပါင္းထားတဲ့ အစုအေ၀းၾကီးတစ္ခုသာလွ်င္ၿဖစ္တယ္ အဲ့ object ေတြထဲမွာမွ data နဲ႔ behavior တို႔ပါ၀င္ပါတယ္။ Object-oriented ဆိုတာက methodology တစ္ခုၿဖစ္ၿပီးေတာ့ သူ႕မွာ concept ေတြရွိတယ္။

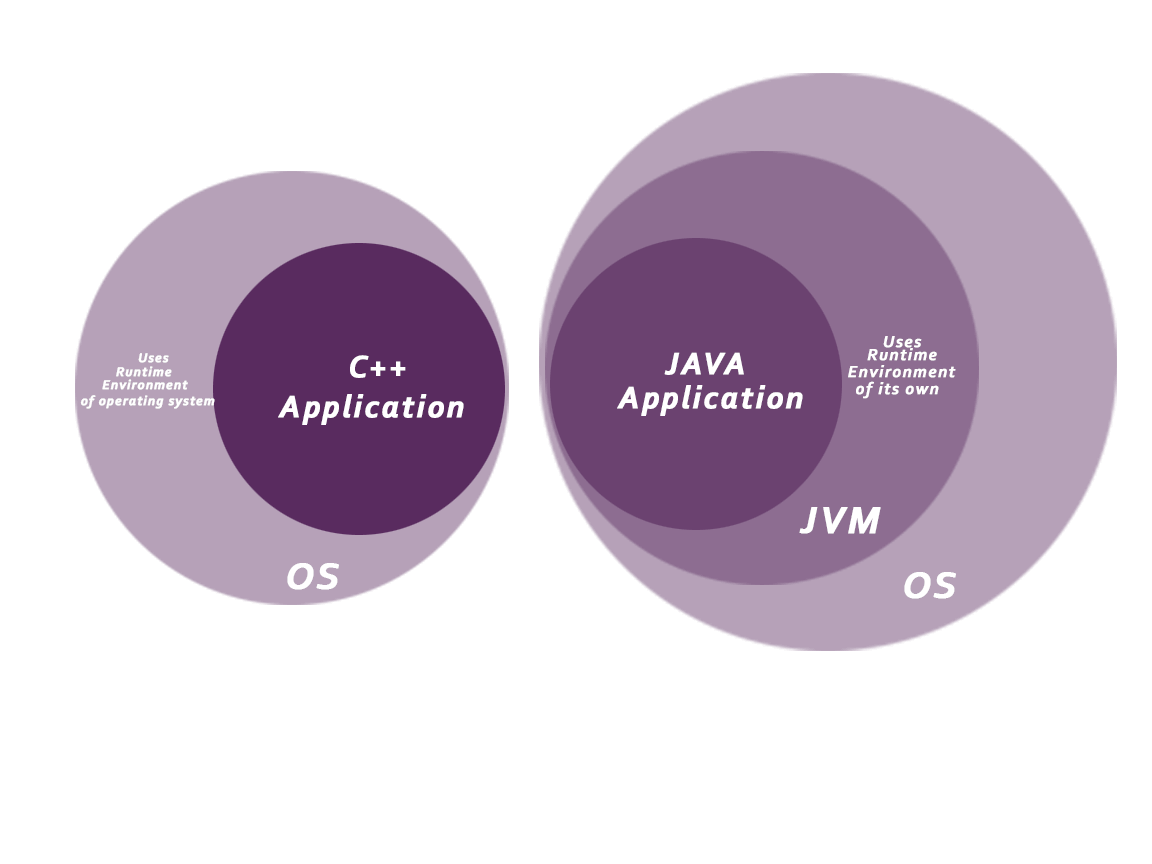
1. [Object](https://www.javatpoint.com/object-and-class-in-java)
2. Class
3. [Inheritance](https://www.javatpoint.com/inheritance-in-java)
4. [Polymorphism](https://www.javatpoint.com/runtime-polymorphism-in-java)
5. [Abstraction](https://www.javatpoint.com/abstract-class-in-java)
6. Encapsulation

**Platform Independence**: မည္သည့္ os ေပၚမဆိုတက္ run လို႔ရလို႔။



**Secured**: java ကို security ေတာ္ေတာ္ေကာင္းတယ္လို႔ သိထားၾကတယ္ ။ဘာလို႔လဲဆိုေတာ့ သူက free virus systmes ကို support ေပးထားတယ္ တစ္ၿခား အခ်က္ေတြကေတာ့

* တိက်ေသခ်ာတဲ့ pointer မရွိဘူး
* **virtual machine sandbox တစ္ခုအတြင္းမွာ run တယ္**



**Classloader**: သူက java ရဲ႕ အစိတ္အပိုင္းတစ္ခုၿဖစ္တယ္။သူ႕က java classes ေတြကို Java Virtual Machine ၾကီးေပၚကို dynamically ဆြဲတင္ဖို႔အတြက္ အသံုးၿပဳၿခင္းၿဖစ္တယ္။

**Bytecode Verifier**: code ေတြကို legal ၿဖစ္မၿဖစ္ စစ္ေဆးၿပီး object မ်ားကို access လုပ္ခြင့္ေပးမေပးကို ဆံုးၿဖတ္တယ္။

**Security Manager**: ဘယ္ class ေတြက local disk ေပၚမွာ reading သို႔မဟုတ္ writing လုပ္မလဲဆိုတာကို ဆံုးၿဖတ္ေပးတယ္။

Java က default အေနနဲ႔ security ၃ ခုကို လုပ္ေဆာင္ေပးထားတယ္။ဘာေတြလဲဆိုေတာ့ SSL, JAAS and Cryptography တို႔ၿဖစ္ၾကတယ္

**Robust:** Robust ဆိုတာ strong ၿဖစ္တာကို ဆိုလိုၿဖင္းၿဖစ္တယ္။ Java က robust ၿဖစ္တယ္။

1. သူက Memory management မွာ အလြန္ေကာင္းတယ္။
2. တိက်တဲ့ Pointer မရွိတဲ့အတြက္ေၾကာင့္ security ကိုပိုမိုေကာင္းမြန္ေစတယ္။
3. JVM ေပၚမွာ automatic garbage collection လုပ္ေဆာင္ႏိုင္တယ္။
4. Java မွာ exception handling နဲ႔ type checking mechanism လုပ္ေဆာင္ႏိုင္တယ္။

အဲ့တာေတြအကုန္လံုးက Robust လို႔ေခၚဆိုႏိုင္တယ္

**Architecture-neutral**: C Language မွာဆိုရင္ 2 bytes ဆိုရင္ 32 bit ၿဖစ္ၿပီးေတာ့ 4 bytes ဆိုရင္ 64 bit ၿဖစ္တယ္။

Java မွာဆိုရင္ေတာ့ 32 ၿဖစ္ၿဖစ္ 64 ၿဖစ္ၿဖစ္ 4 bytes ပဲ။

**Portable**: Java မွာက default အေနနဲ႔ java bytecode အေနနဲ႔သယ္ေဆာင္သြားတဲ့အတြက္ ဘယ္ platform မွာမဆို run ႏိုင္တယ္။ဘယ္ implementation မွလုပ္ေဆာင္စရာမလို။

**Multi-threaded**: thread ဆိုတာက သီးၿခားရပ္တည္ေနတဲ့ program အေသးေသးေလးတစ္ခုပဲ။ သူတို႔ေတြက တစ္ၿပိဳင္တည္း execute လုပ္ေဆာင္ႏိုင္တယ္။ tasks ေတြအမ်ားၾကီးကို တစ္ၿပိဳင္တည္းမွာ လုပ္ေဆာင္ႏိုင္တာကို multi-thread လို႔ေခၚတယ္။သူက ဘာေကာင္းတာလဲဆိုေတာ့ thread တစ္ခုခ်င္းစီက memory ကို မပိုင္ဆိုင္ဘူး သူတို႔က memory ကို common memory area ကေန shares သံုးၾကတာၿဖစ္တယ္။Threads ေတြက Web Application တို႔ multi-media တို႔ အတြက္ ေတာ္ေတာ္ကို အေရးပါပါတယ္။

**Dynamic**: Java က dynamic language ၿဖစ္တယ္။ဘာလို႔ေၿပာရလဲဆိုေတာ့ class file ေတြကို dynamic ဆြဲတင္ေပးလို႔ dynamic ၿဖစ္တယ္လို႔ ေၿပာတာ။

ေနာက္ဆံုးတစ္ခ်က္အေနနဲ႔ Java က dynamic compilation နဲ႔ automatic memory management (garbage collection) ကို လုပ္ေဆာင္ေပးပါတယ္။

**First Java Program**

Java program တစ္ပုဒ္မေရးခင္ ၿပဳလုပ္ရမွာက

1. JDK install လုပ္ရမယ္။
2. Environment setup variables ထည့္ရမယ္။

Environment setup variable ကိုဘယ္လိုထည့္သြင္းမလဲ

သူ႕မွာ ၂ မ်ိဳး ရွိတယ္။ temporary နဲ႔ permanent ဆိုၿပီးေတာ့ေပါ့။

1. My Computer ကို right click ေခါက္ properties ကို ႏွိပ္
2. Advanced System Settings ကို ႏွိပ္ System Properties dialog တက္လာရင္ Advanced tag ကိုသြား ေရာက္ရင္ environment variables ကို ႏွိပ္
3. “Environment Variables” dialog box တက္လာရင္ ေအာက္တစ္ပိုင္းက(System Variables) New button ကို ႏွိပ္။ရွိၿပီးသားဆိုရင္ေတာ့ Edit Button ႏွိပ္ေပါ့။
   1. (Java\_Homeအတြက္) “New User Variable” ဆိုတဲ့ dialog box ေလးတက္လာလိမ့္မယ္။ variable name မွာ JAVA\_HOME လို႔ရိုက္မယ္။variable value မွာ ေရွ႕က install လုပ္ထားတဲ့ jdk folder ထိ path လမ္းေၾကာင္း(C:\Program Files\Java\jdk1.8.0\_45)ကို copy ကူးၿပီး ထည့္ေပးလိုက္ပါ။
   2. (Path အတြက္) “New User Variable” ဆိုတဲ့ dialog box ေလးတက္လာလိမ့္မယ္။ variable name မွာ Path လို႔ရိုက္မယ္။variable value မွာ ေရွ႕က install လုပ္ထားတဲ့ bin folder ထိ path လမ္းေၾကာင္း(C:\Program Files\Java\jdk1.8.0\_45\bin;)ကို copy ကူးၿပီး ထည့္ေပးလိုက္ပါ။(ေနာက္ဆံုးမွာ “;” ပါကိုပါရမယ္)
4. ၿပီးသြားရင္ေတာ့ next button မ်ားကို ႏွိပ္ၿပီး ထြက္လိုက္ရံုပဲ။

# JVM (Java Virtual Machine) Architecture

JVM ဆိုတာက abstract machine တစ္ခုၿဖစ္ၿပီး runtime environment တစ္ခုလည္း ဟုတ္တယ္။သူ႕ထဲမွာ java byte code ေတြကို execute လုပ္ႏိုင္တယ္။

JVM က hardware software အမ်ားၾကီးထဲမွာ အဆင္ေၿပေၿပလုပ္ေဆာင္ႏိုင္တယ္ ဆိုလိုတာက သူက platform တစ္ခုခုေပၚမွာ run ရမယ္ဆိုတဲ့ သေဘာေပါ့။

**JVM ဆိုတာ ဘာလဲ**

* 1. Specification
  2. An Implementation: သူ႕ရဲ႕ implementation က JRE ၿဖစ္တယ္။