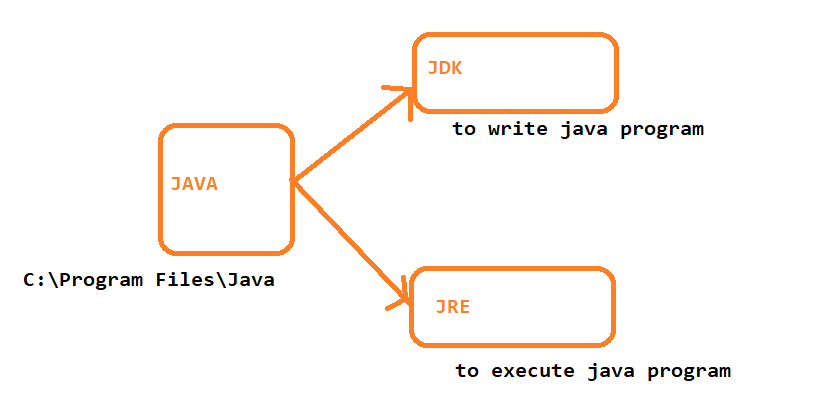
Java

* Brush-up on Java concepts
  + OOPS Concepts
    - Classes
    - Objects
    - Abstraction
    - Encapsulation
    - polymorphism
    - Inhertence
  + SIB and IIB
  + Interfaces
  + Access Specifiers
  + Access Modifiers / Non functional Access Specifiers
  + Abstract Class
  + packages
  + import statements
  + data types – types
  + variables
  + methods
  + loops – types of loops
  + conditional statements – types of conditional statements
  + **String – Class**
  + **File –Reading and writing in JAVA**
  + Exceptions how to handle Exceptions
  + **Collections**
  + **Generics**
* Java-1.8
  + Inner Class – Advanced Java
  + Interfaces with Advanced Features
  + forEach & Consumer Class
  + Lambda Expressions
  + **:: Operator**
  + Stream APIs
  + Improvements made to Collection API
  + Improvements made to IO package
  + Date / Time APIs

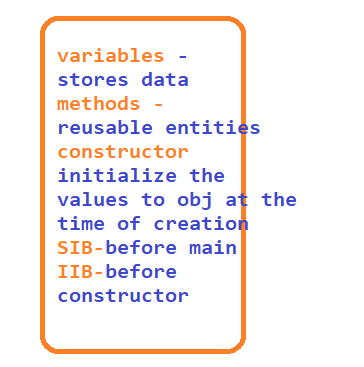
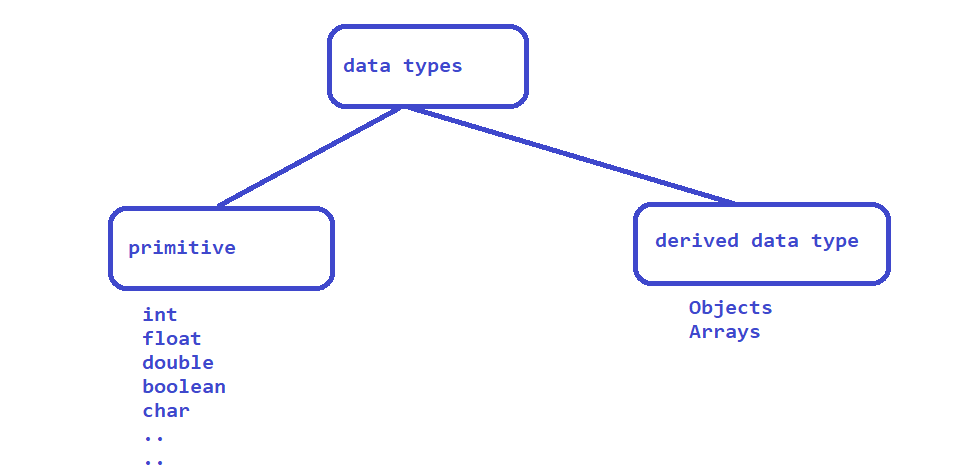
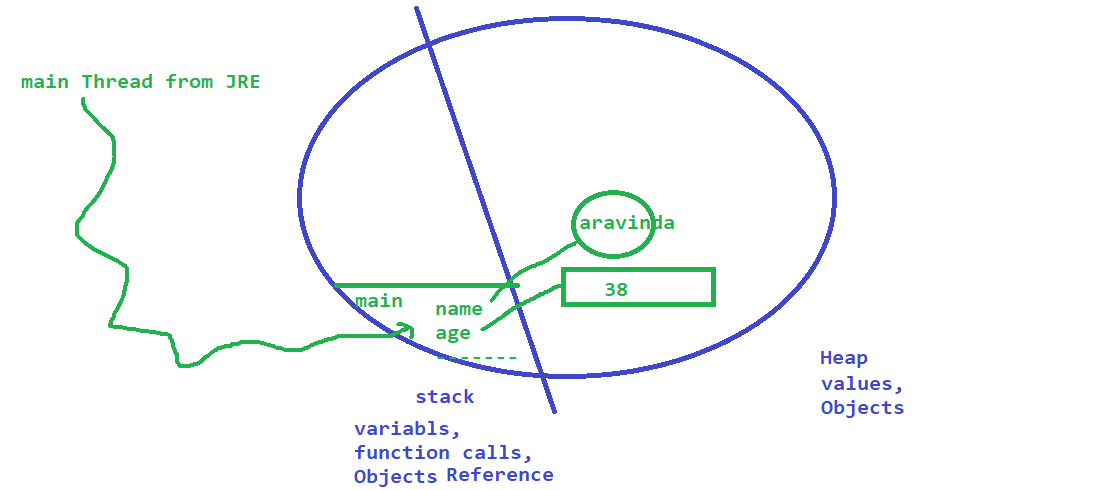
Introduction to Software and libs



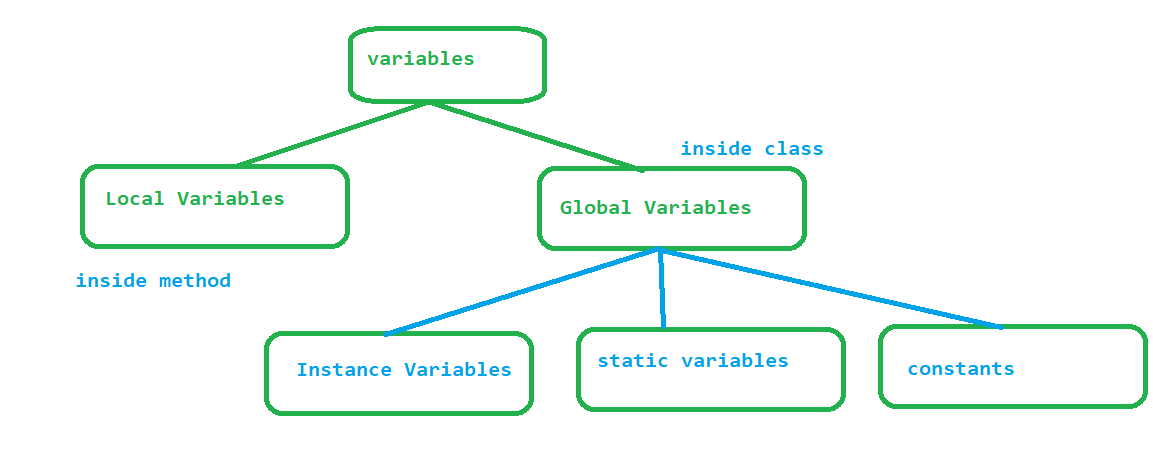
C:\Program Files\Java\jdk1.8.0\_201\bin

C:\Program Files\Java\jdk1.8.0\_201\lib

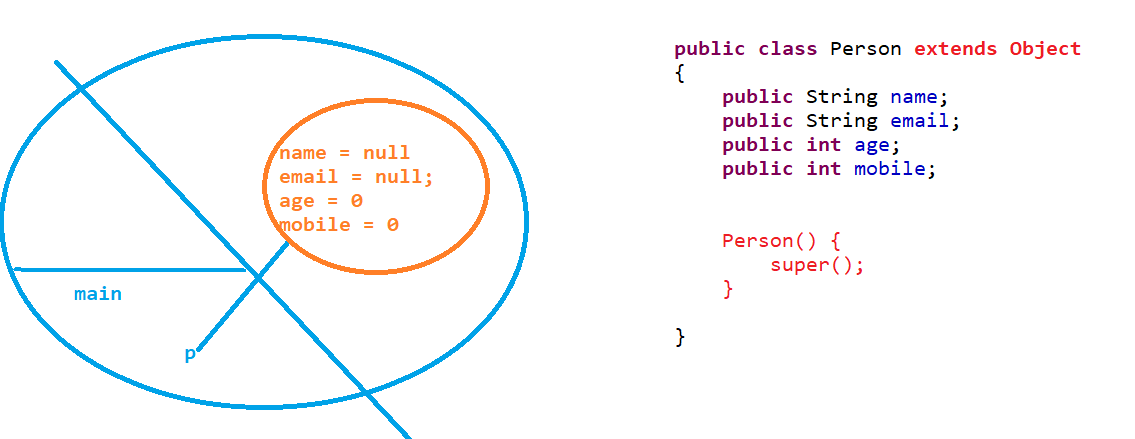
Oops Concepts

* Class
* 
* Object
* package
  + 
* Data types
* 
* 

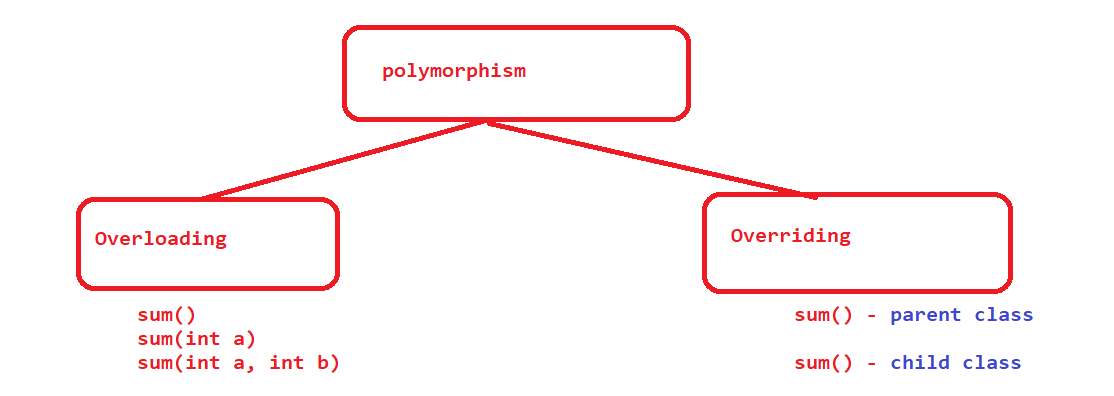
Variables



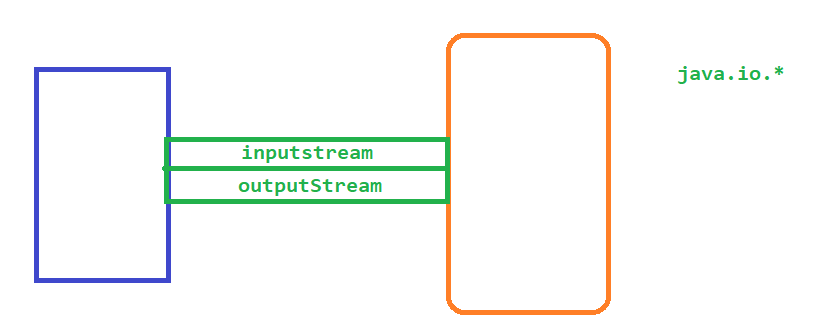
Constructor



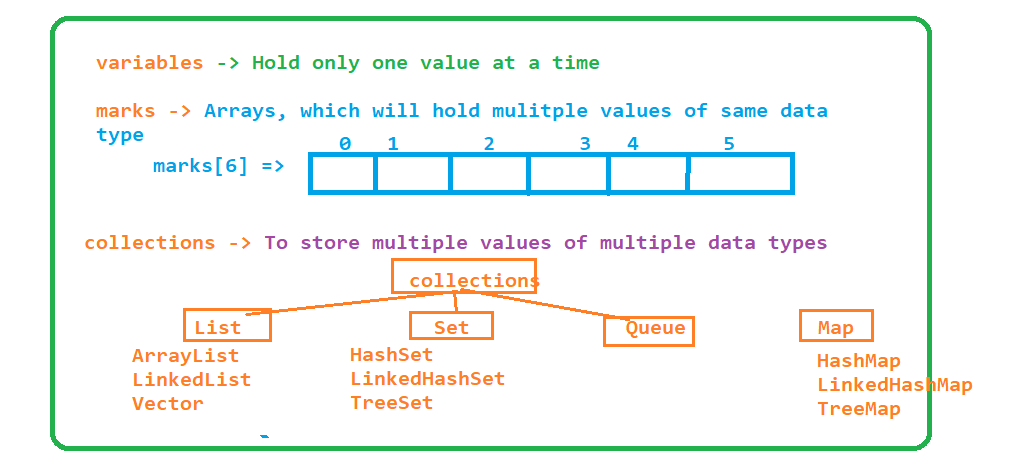
Polymorphism



Files

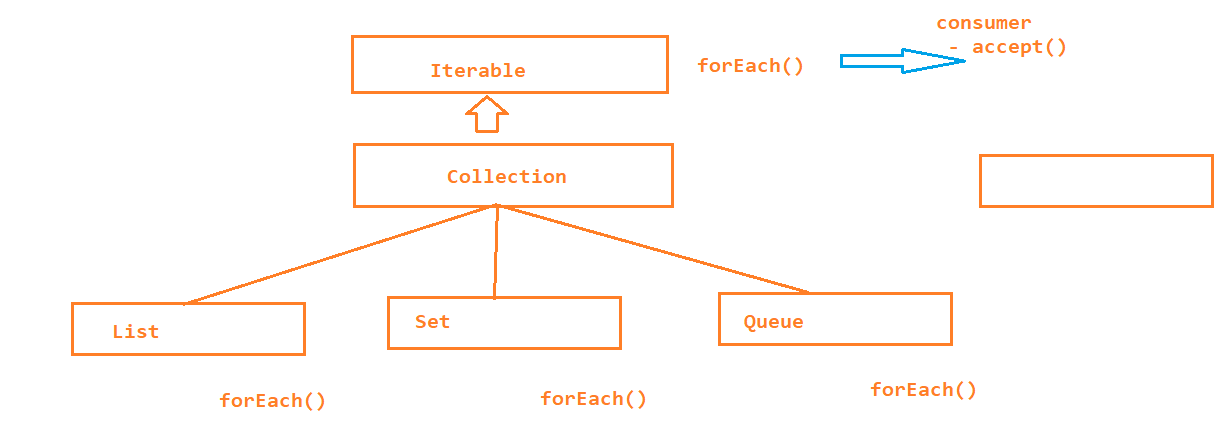


Collections – java.util.\*



Java-1.8

## forEach



\* Consumer is a interface introduced from java 1.8 which has an unimplemented method accept

* forEach method is also introduced from java1.8 which is present inside Iterable interface
* Iterable is a parent for all the collection classes
* Foreach expect object to Consumer class
* to Foreach loop we have to implement the actual logic inside the accept method of consumer class

1. Inner Class

Inner classes are also called as Nested Classes, mainly used to achieve

1. Better Encapsulation

2. To group the logically similar classes

3. for better readable and maintainable code

Types :

1. static inner class

* is a member of parent class
* it is accessible from all the static members of the parent class

2. local inner class

* it is a member of a method
* local inner classes are accessible only inside the method, not from any other member of a class
* LIC are local to the method only

3. Anonymous inner class

* are the class without any name
* we can create object to interface using anonymous inner classes

1. Lambda Expressions

Lambda expressions are the block of code, which can be executed without object

Lambda expressions are similar to methods ( advanced methods )

Faster execution – because it executes without object

While writing lambda expression

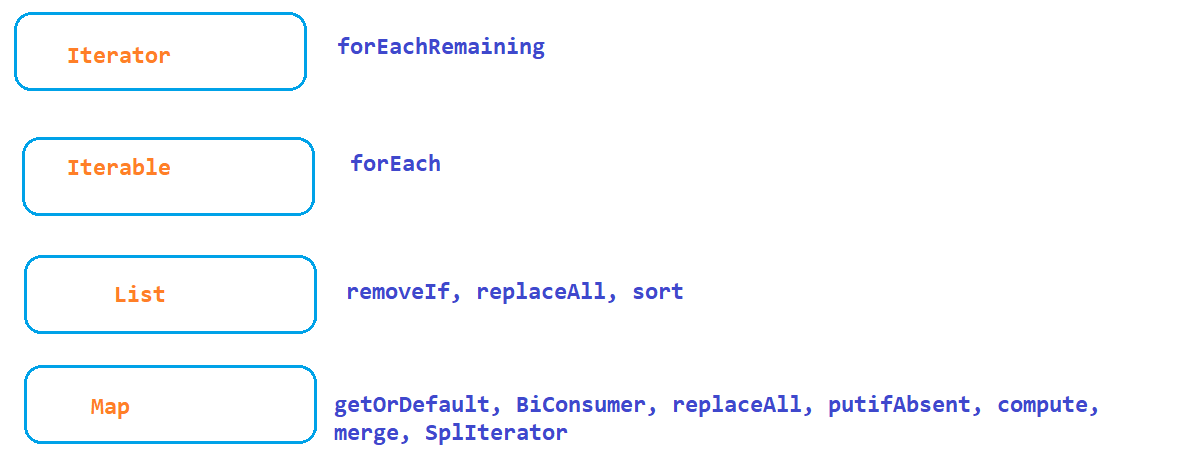
* Lambda expressions with arguments
  + Data type is optional when we are implementing method using lambda expression
* Lambda expressions without arguments
* Lambda expressions with return type
* Lambda expressions without return type
  + if we have one line implementation then return keyword is not needed
  + return keyword is needed if there is multiple line of implementation
* Lambda expressions for forEach loop

:: Operator

:: is used to assign the existing implementation to the unimplemented method of a Interface

* Unimplemented method of an interface can be implemented by
  + Class
  + Annonymous inner class
  + Lambda expression etc

Collections



IO Improvements

* list – list all the files of the current folder / dir, it will not go to subdir
* walk - – list all the files of the current folder as well as in the sub dir till the last file is reached
* lines – will read and print each and every line in a file

Steam

* in java.util package there is a sub package called stream introduced in 1.8 mainly to process the data
* Processing group of elements that are in array or Collections can be easily done here.
* Processing - > finding max, min, avg, count, etc

Date and Time

* Thread safety
* Difficult to implement date and time with it comes to different time zones
* Very Complicated before 1.8
  + LocalDate
  + LocalTiime
  + LocalDateTime