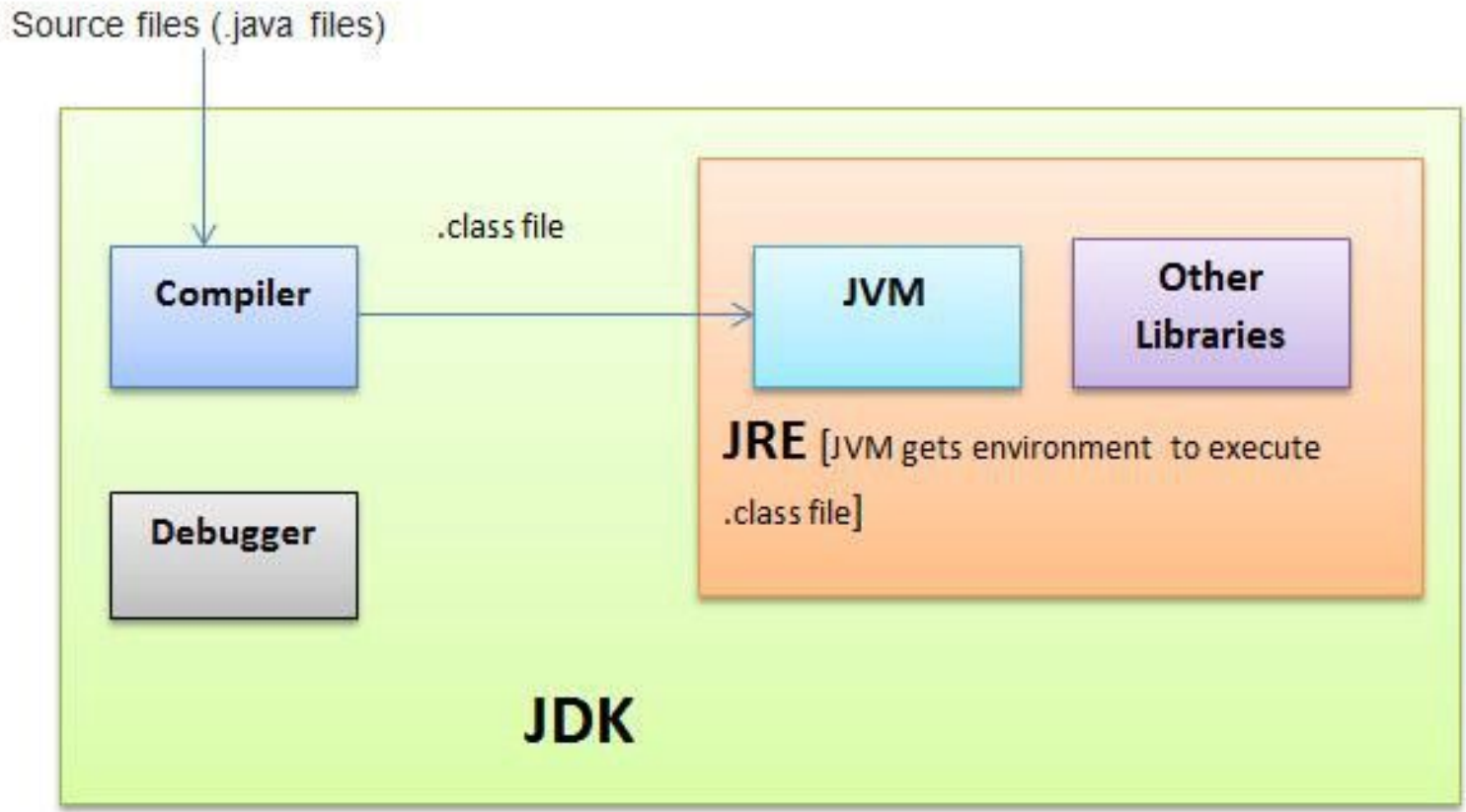
JAVA

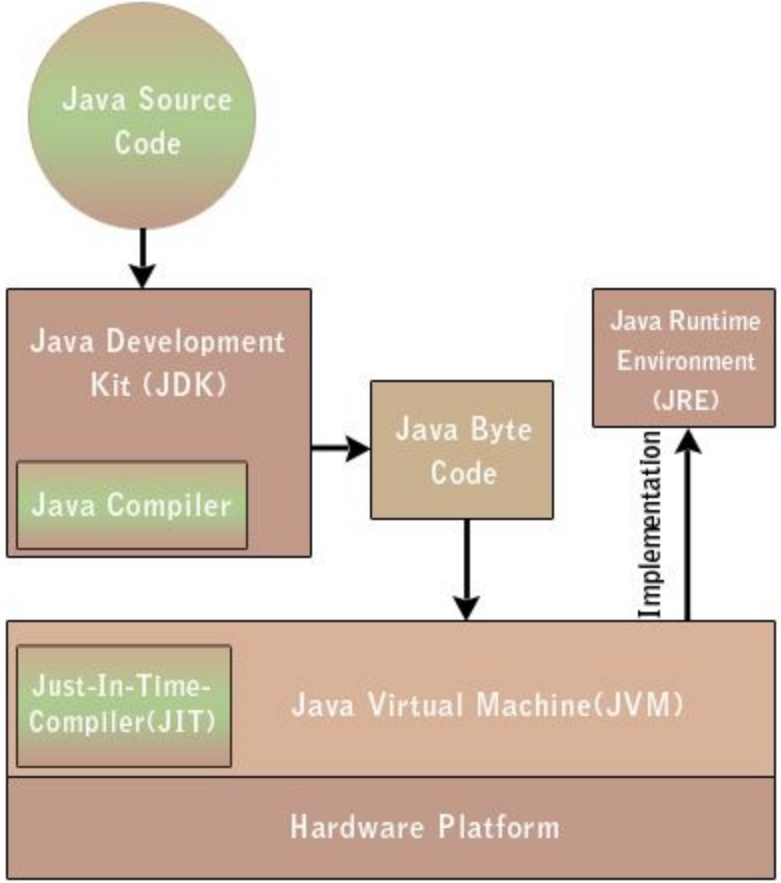
Divyashree B A

Install JDK

- Open Command prompt and check if java installed
- Java -version
- If not installed install JDK(java) and set system variables.
- New variable JAVA_HOME = path upto jdk
- Edit path variable and add “path upto bin”

Install Eclipse


Source files (.java files)
.class file
Compiler
JVM
Other Libraries
JRE [JVM gets environment to execute .class file]
Debugger
JDK


Java Source Code
Java Development Kit (JDK)
Java Compiler
Java Byte Code
Java Runtime Environment (JRE)
Implementation
Just-In-Time-Compiler(JIT)
Java Virtual Machine(JVM)
Hardware Platform

Variable: which holds the value.

Syntax:

```
dataType nameOfVariable;              //declaration
nameOfVariable= valueOfVariable    // initialization
```

```
dataType nameOfVariable=valueOfVariable;  //declaration + initialization
```

Types of variable:

- Local variable: defined locally inside method body.
  - Note: scope of the variable will be accessed to that method only.
- Instance variable: defined globally, outside the method body (inside the class).
  - Note: scope of the variable will be accessed by all methods of same class
  - as well as other classes based on access modifiers.
- Class/Static variables
  - Declared at the class level.(static)

Data Type:

- Primitive data type
  - Stores the value
  - Pre defined by programming language.
  - No additional methods
  - byte, short, int, long, float, double, char, boolean
- Non-primitive data type
  - Stores the reference which refers memory location where data gets stored
  - Created by programers
  - Array, String, classes

Conditional statement:

- if
- if else
- Nested if else
- Switch

Loop:

- while
- do while
- For
- Enhanced for loop(for each loop)

Loop Control Statements

- Break
- Continue

Scanner class in java

Functions

No parameter No return type

With Parameter and return type

Function overloading

Strings in java

There are two ways to create a string in Java:

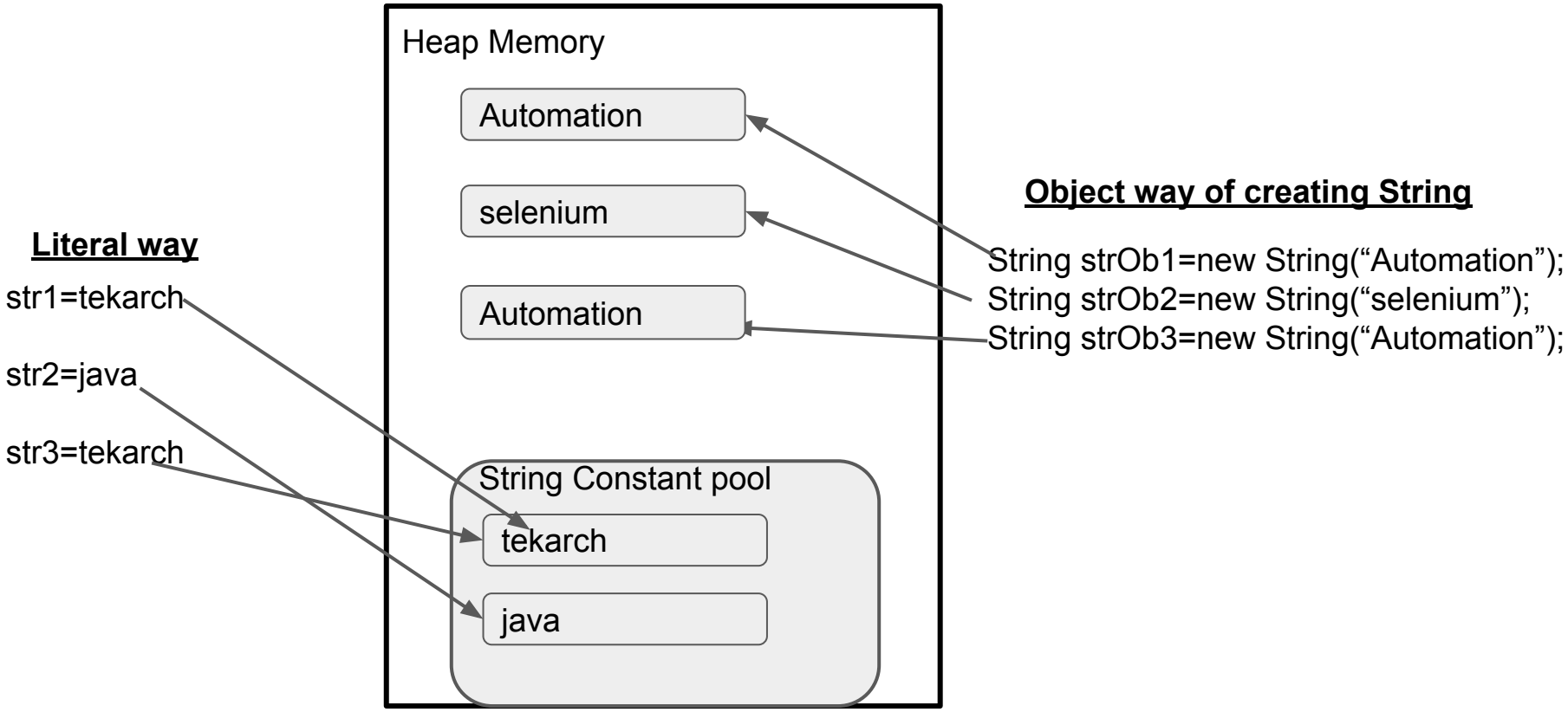
1. String literal

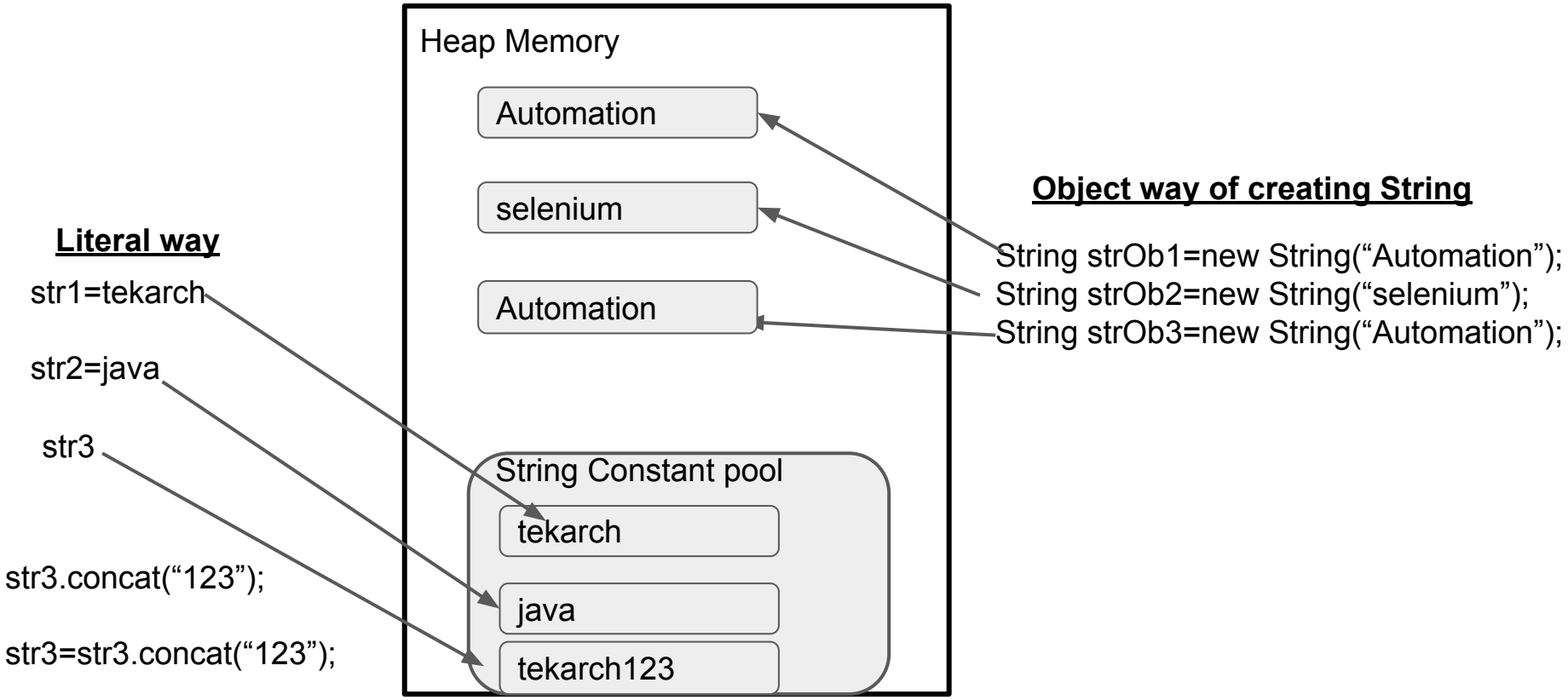
   String s = “tekarch”;
   Whenever a String Object is created as a literal, the object will be created in String constant pool.

2. Using new keyword

   String s = new String (“tekarch”);
   dynamically allocated. In case of String are dynamically allocated they are assigned a new memory location in heap

Strings are immutable in JAVA


Heap Memory
Automation
selenium
Automation
String Constant pool
tekarch
java
Literal way
str1=tekarch
str2=java
str3=tekarch
Object way of creating String
String strOb1=new String(“Automation”);
String strOb2=new String(“selenium”);
String strOb3=new String(“Automation”);


Heap Memory
Automation
selenium
Automation
String Constant pool
tekarch
java
tekarch123
Literal way
str1=tekarch
str2=java
str3
str3.concat("123");
str3=str3.concat("123");
Object way of creating String
String strOb1=new String("Automation");
String strOb2=new String("selenium");
String strOb3=new String("Automation");

Mutable classes

StringBuffer:

Syntax:

```
StringBuffer s = new StringBuffer("GeeksforGeeks");
```

StringBuilder:

Syntax:

```
StringBuilder str = new StringBuilder();

str.append("GFG");
```

Strings comparisons

- equals()
- ==
- compareTo()

Arrays in java

- Single dimensional
  - Int[] a=new int[5];
- Two dimensional
  - Int[][] a=new int[5][3];

Array Utility class in java

Wrapper classes class representation of primitive types

- Integer,Character,Float,Boolean

- Boxing
  - Integer ob=new Integer(k); // where int k=10;
- Unboxing
  - syso(ob.intValue)
- Auto boxing
  - Integer ob=k;
- Auto unboxing
  - syso(ob)

Class and object

- Attributes
  - Instant variables
  - class/static variables
- Functions
- constructors

  A constructor in Java is a **special method** that is used to initialize objects. The constructor is called when an object of a class is created. It can be used to set initial values for object attributes

- toString() method

  If you want to represent any object as a string, **toString() method** comes into existence.
  The toString() method returns the String representation of the object.

- This keyword
- Super keyword

Array of Objects

Static keyword in Java

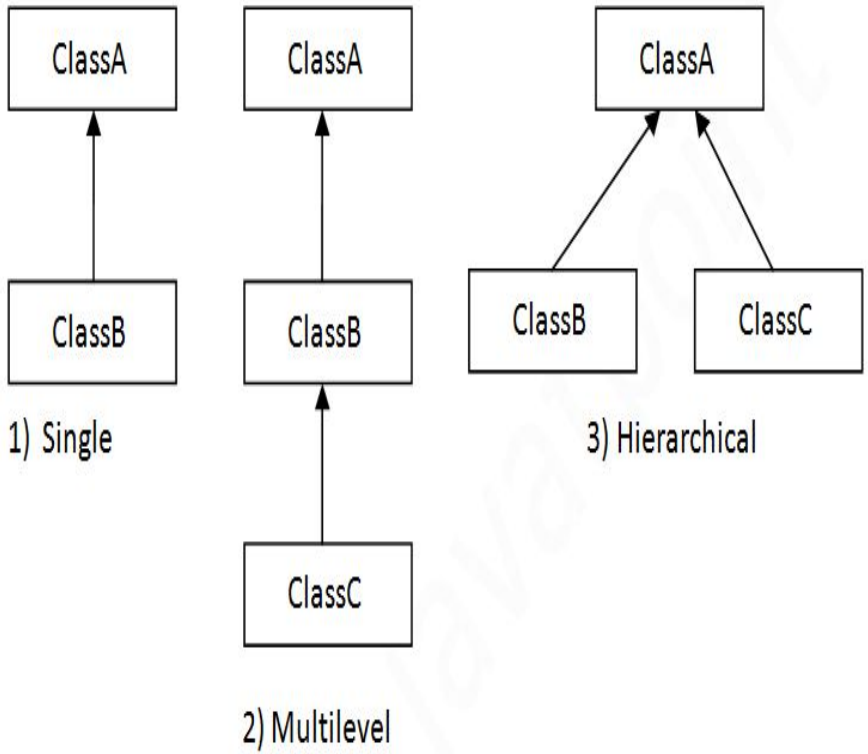
- Blocks
  - static block that gets executed exactly once, when the class is first loaded
- Variables
  - When a variable is declared as static, then a single copy of variable is created and shared among all objects at class level.(Global variable)
- Methods
  - The most common example of a static method is *main( )* method.As discussed above, Any static member can be accessed before any objects of its class are created, and without reference to any object.
- nested classe

OOPs concept:

Inheritance

Inheritance is an important pillar of OOP(Object-Oriented Programming). It is the mechanism in java by which one class is allowed to inherit the features(fields and methods) of another class.

- Single inheritance
- Multi level inheritance
- Hierarchical inheritance
- Multiple inheritance with classes in java is not possible.



Private and static methods cannot overriden. If you do it consider as new method in child class.

Final methods cannot be overriden

Overriding method can allow more access but not less. Protected parent class method can be overriden by public or protected child class method but not private method

Modifiers in JAVA

Access Control Modifiers

Java provides a number of access modifiers to set access levels for classes, variables, methods and constructors. The four access levels are −

- Default
- Private
- Protected
- Public

	default	private	protected	public
Same Class	Yes	Yes	Yes	Yes
Same package subclass	Yes	No	Yes	Yes
Same package non-subclass	Yes	No	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes

Non-Access Modifiers

Java provides a number of non-access modifiers to achieve many other functionality.

- Static
- Final
- Abstarct
- synchronized and volatile (which are used for threads).

- Final
- Finally
- Finalize()

Oops Concepts

- Encapsulation
  - Encapsulation in Java is **a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit**.
- Data hiding
  - **Data hiding** restricts the data use to assure data security.
- Polymorphism
  - Polymorphism in Java is **the ability of an object to take many forms**
  - Run time(overriding)(occurs when a derived class has a definition for one of the member functions of the base class. That base function is said to be **overridden**)
  - Compile time (overloading)(occurs when there are multiple functions with same name but different parameter styles)
- Inheritance
- Abstraction
  - Data **abstraction** is the process of hiding certain details and showing only essential information to the user.
  - In java, abstraction is achieved by interfaces and abstract classes. We can achieve 100% abstraction using interfaces.

- Abstract class
  - Class with abstract keyword
  - May or may not have abstract methods(it does not have a body).An abstract class can have both abstract and regular methods
  - If atleast one method is abstract then class should be abstract class
  - Extended class should implement all abstract methods
  - If an abstract class extending other abstract class then child class no need to implement the abstract methods in the parent class
  - Abstract class variables may be static non static final non final
  - We cannot create object an any abstract class

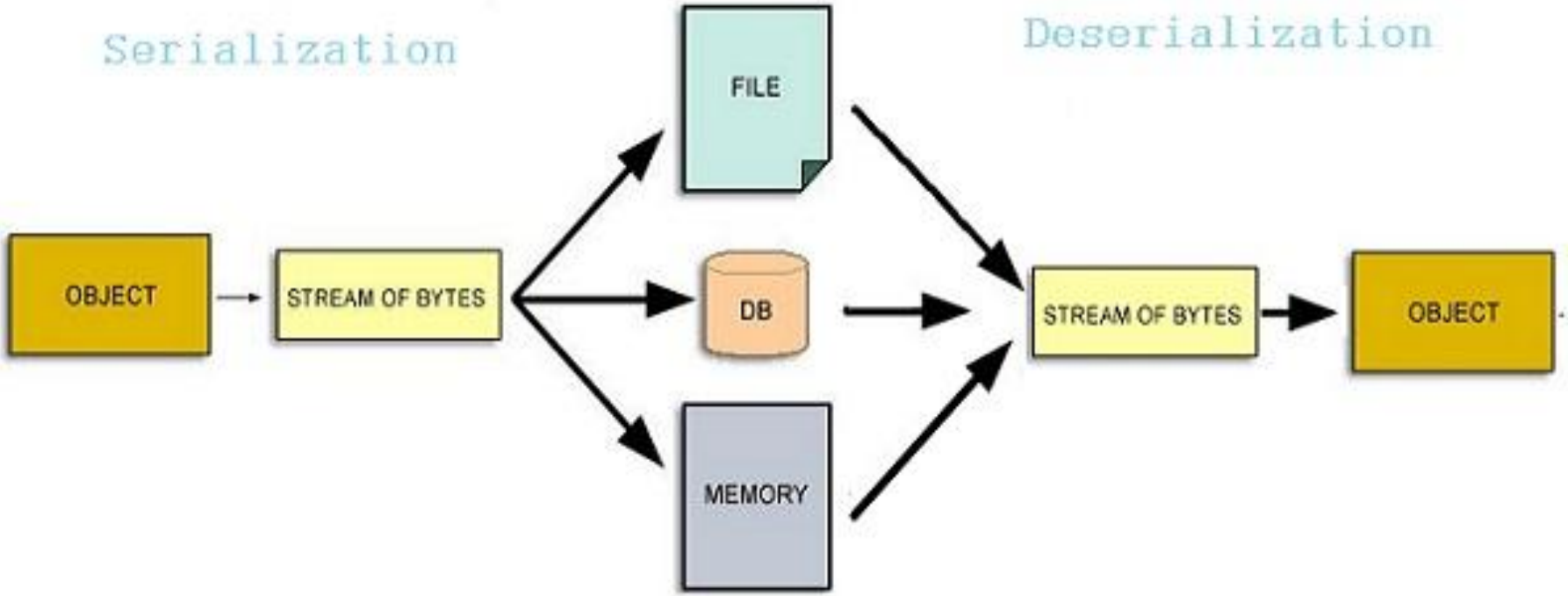
Interface

- To declare an interface, use **interface** keyword
- Like a class, an interface can have methods and variables, but the methods declared in an interface are by default abstract (only method signature, no body).
- Variables of interface is final and static by default.
- A class that implements an interface must implement all the methods declared in the interface. To implement interface use **implements** keyword.

Singleton class

- a singleton class is a class that can have only one object (an instance of the class) at a time.
  - Make constructor as private.
  - Write a static method that has return type object of this singleton class

Serialization and deserialization


Serialization
Deserialization
FILE
OBJECT
STREAM OF BYTES
DB
STREAM OF BYTES
OBJECT
MEMORY

Exception

An event that occurs during the execution of a program that disrupts the normal flow of instructions is called an exception

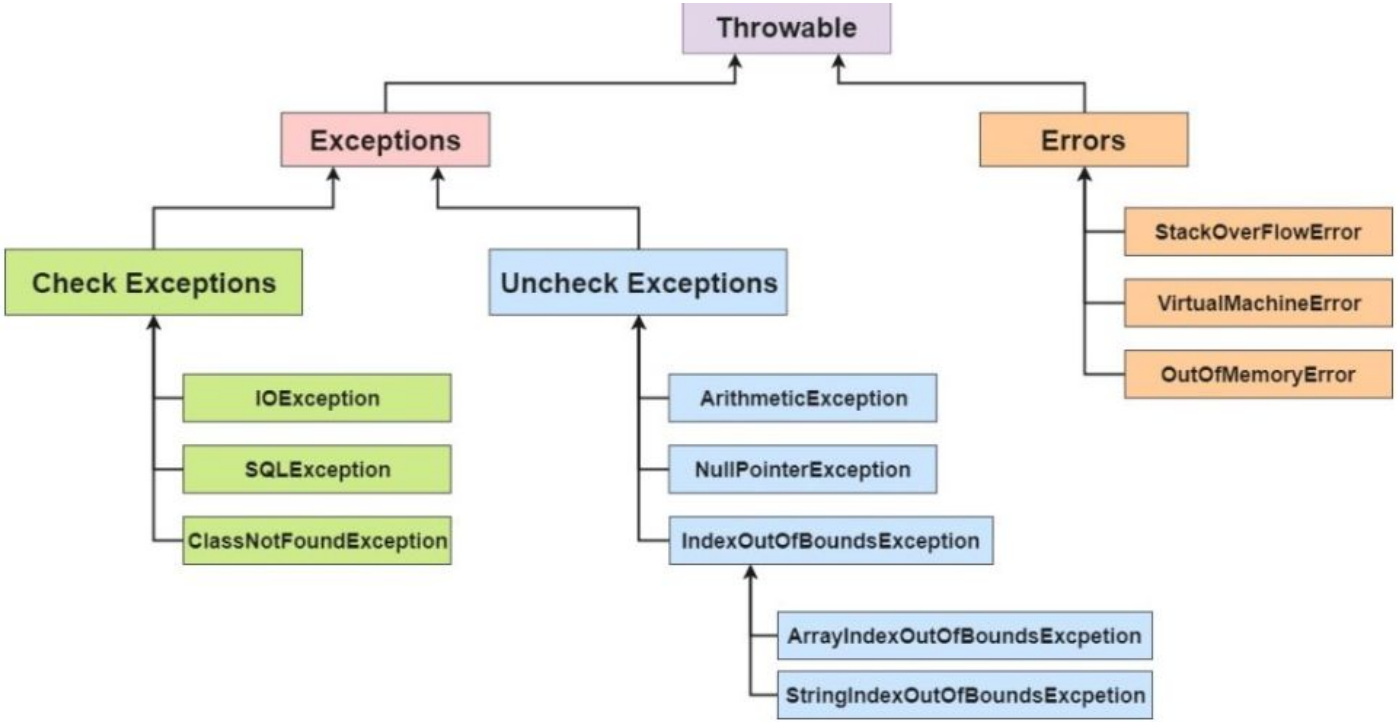
Exceptions can be handled but not Errors

The Exception class is used for exception conditions that the application may need to handle

The Error class is used to indicate a more serious problem in the architecture and should not be handled in the application code.

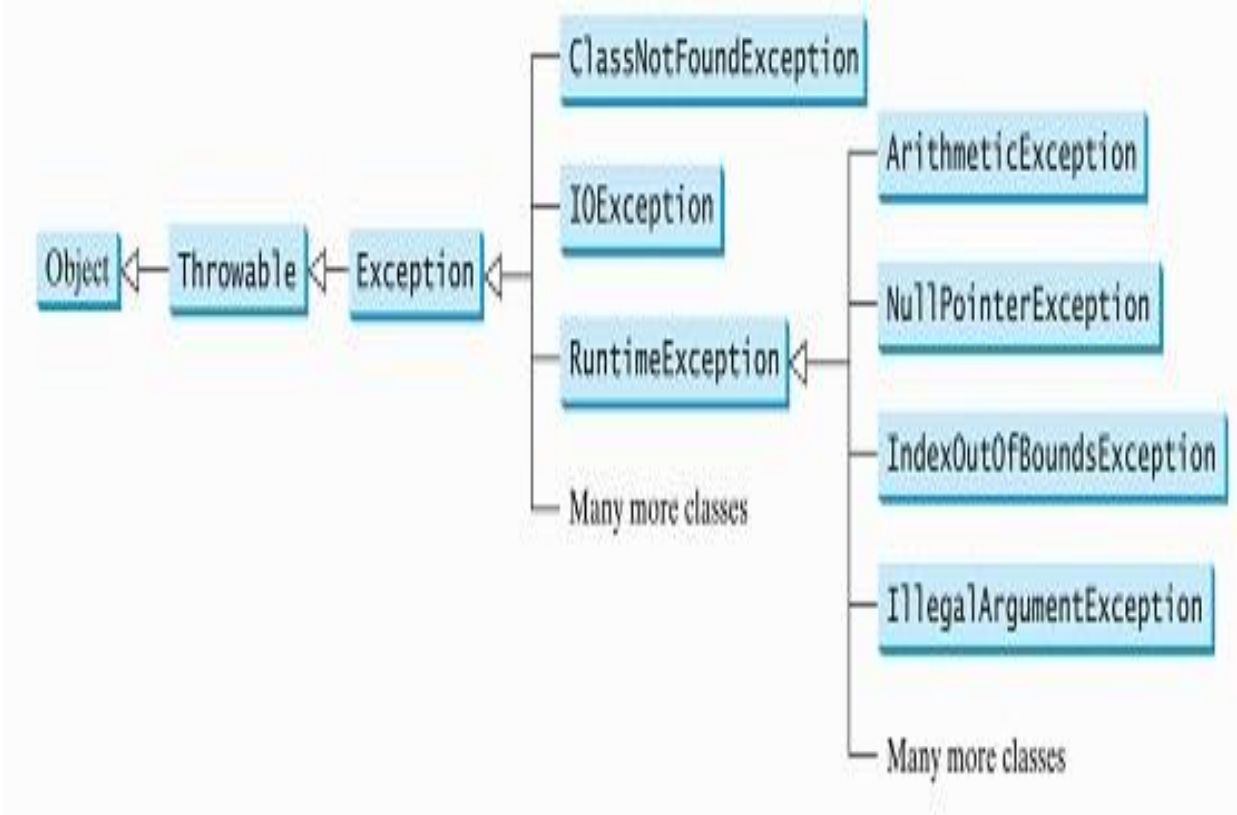
Exception in JAVA

Hierarchy



Built in exception

User defined exception



Abstract Class

A class which contains the **abstract** keyword in its declaration is known as abstract class.

- Abstract classes may or may not contain *abstract methods*, i.e., methods without body (public void get();)
- But, if a class has at least one abstract method, then the class **must** be declared abstract.
- If a class is declared abstract, it cannot be instantiated.
- To use an abstract class, you have to inherit it from another class, provide implementations to the abstract methods in it.
- If you inherit an abstract class, you have to provide implementations to all the abstract methods in it.

Java IO streams

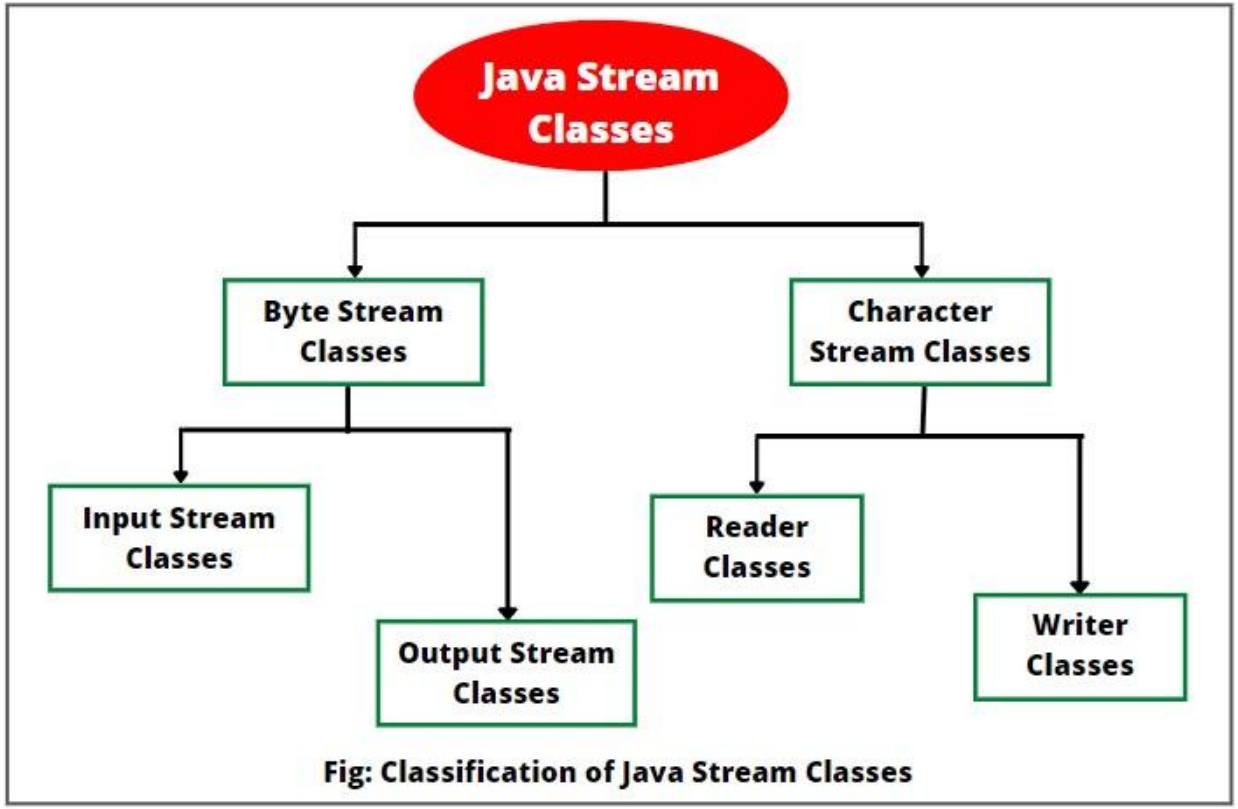
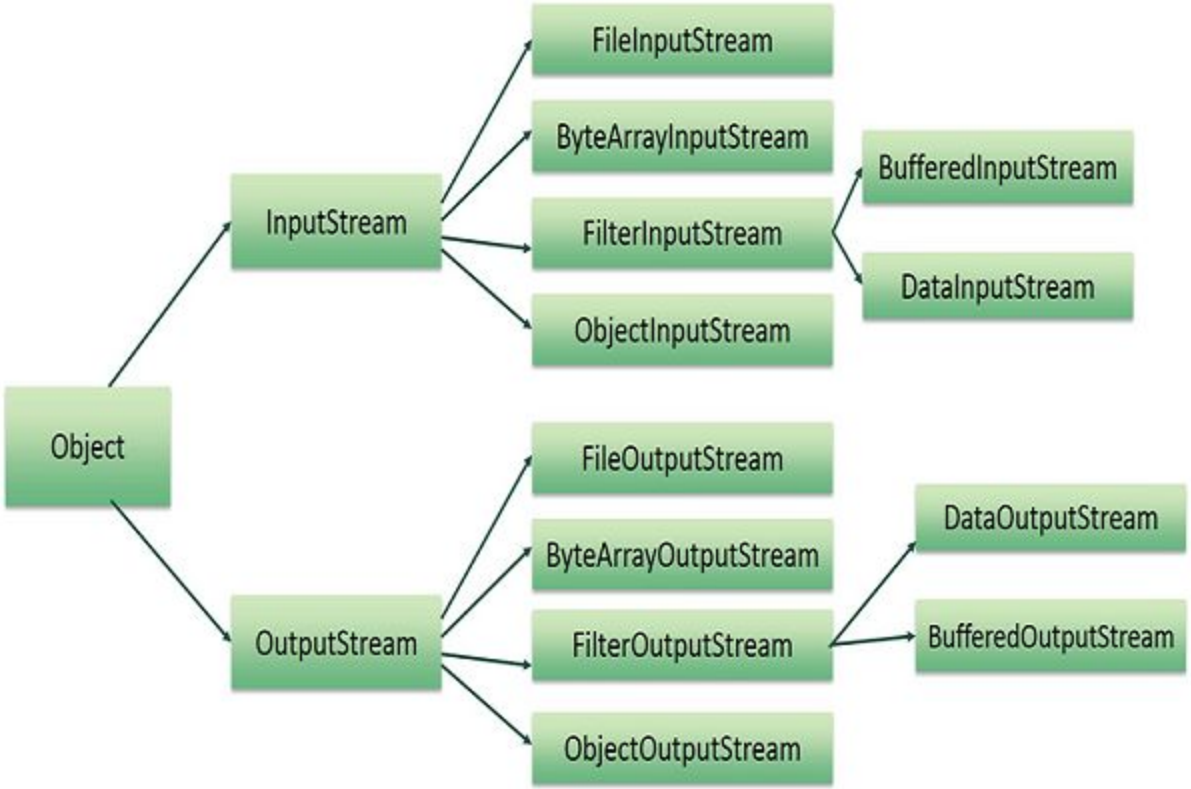
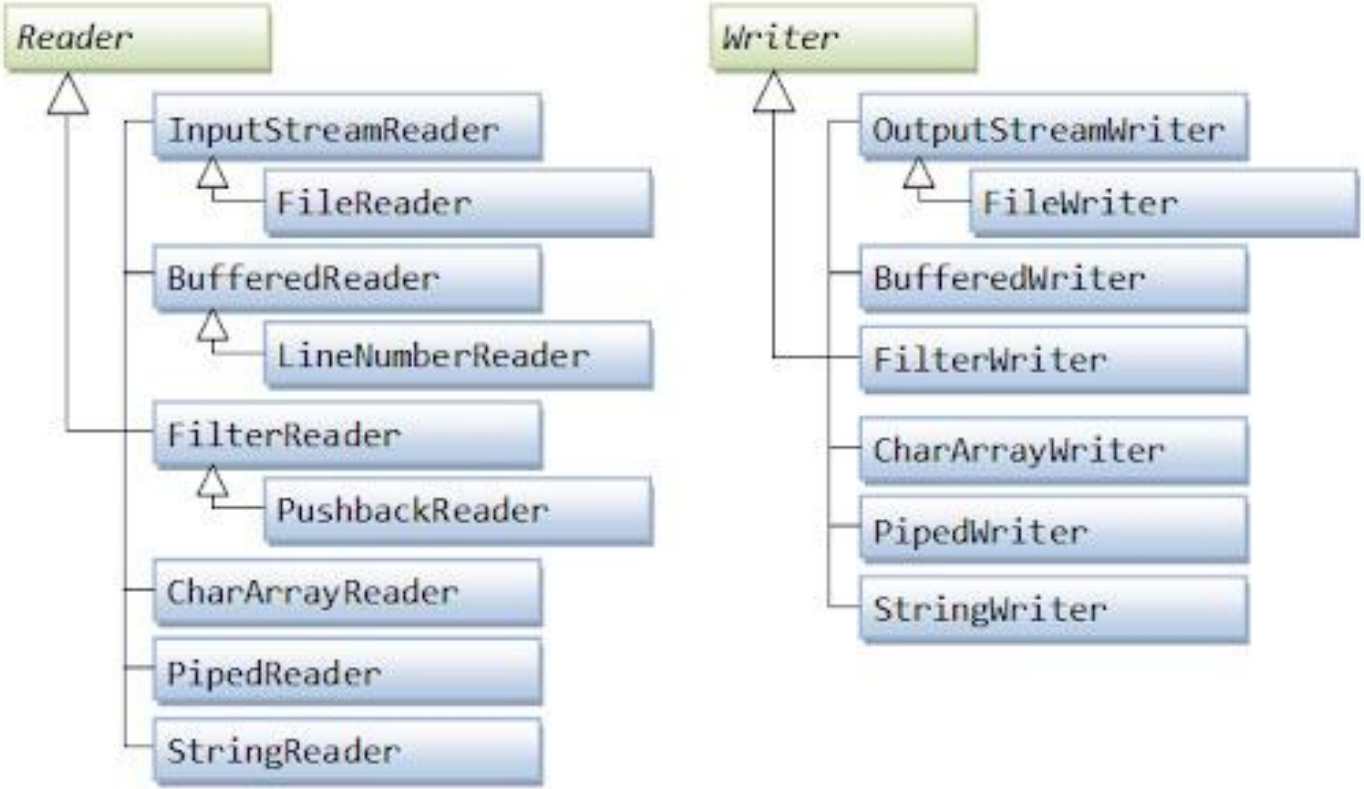


Fig: Classification of Java Stream Classes


Object
InputStream
FileInputStream
ByteArrayInputStream
FilterInputStream
ObjectInputStream
BufferedInputStream
DataInputStream
OutputStream
FileOutputStream
ByteArrayOutputStream
FilterOutputStream
ObjectOutputStream
DataOutputStream
BufferedOutputStream


Reader
InputStreamReader
FileReader
BufferedReader
LineNumberReader
FilterReader
PushbackReader
CharArrayReader
PipedReader
StringReader
Writer
OutputStreamWriter
FileWriter
BufferedWriter
FilterWriter
CharArrayWriter
PipedWriter
StringWriter