



Develop the real time scenarios
based on OOPs concepts and Java
(50 Hours)



In this section, we will discuss:

- Introduction to java, Difference between structural programming and oops programming, OOPs concepts, Encapsulation, Class & Objects, Hello.java, Java environment setup, compilation & execution.
- Java primitive data types including int, float, char, String, Boolean, and double.
- Define Java constants. Declare, assign, and initialize variables.
- Write simple arithmetic statements. Understand operator precedence.
- Understand arithmetic and logical operators.
- Explain the difference between primitive and reference data types
- Use Java decision structures including IF, IF...ELSE, nested IF, and Switch statements.
- Use Java logical operators including AND, OR, and the conditional NOT
- Polymorphism : Static & Dynamic Polymorphism.
- Constructors, Difference between constructors and functions.

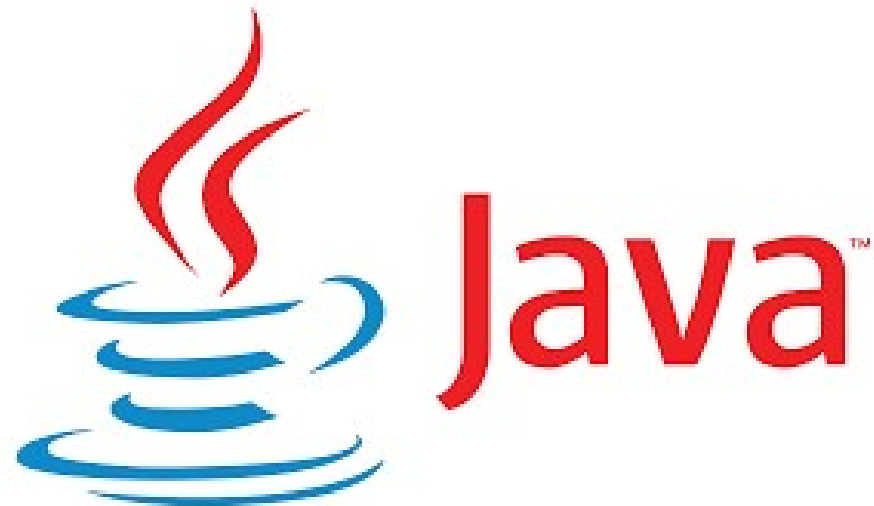


- Declare and initialize a Java array.
- Understand the concepts String manipulation.
- String class & string buffer class.
- Wrapper Classes
- Inheritance: Single level, Multi-level, Multiple.
- Super Class
- Interfaces
- Abstract Classes
- Packages
- Access Specifier
- Inner Classes
- Exception Handling
- Java AWT.
- Event Handling in java.



Introduction to java

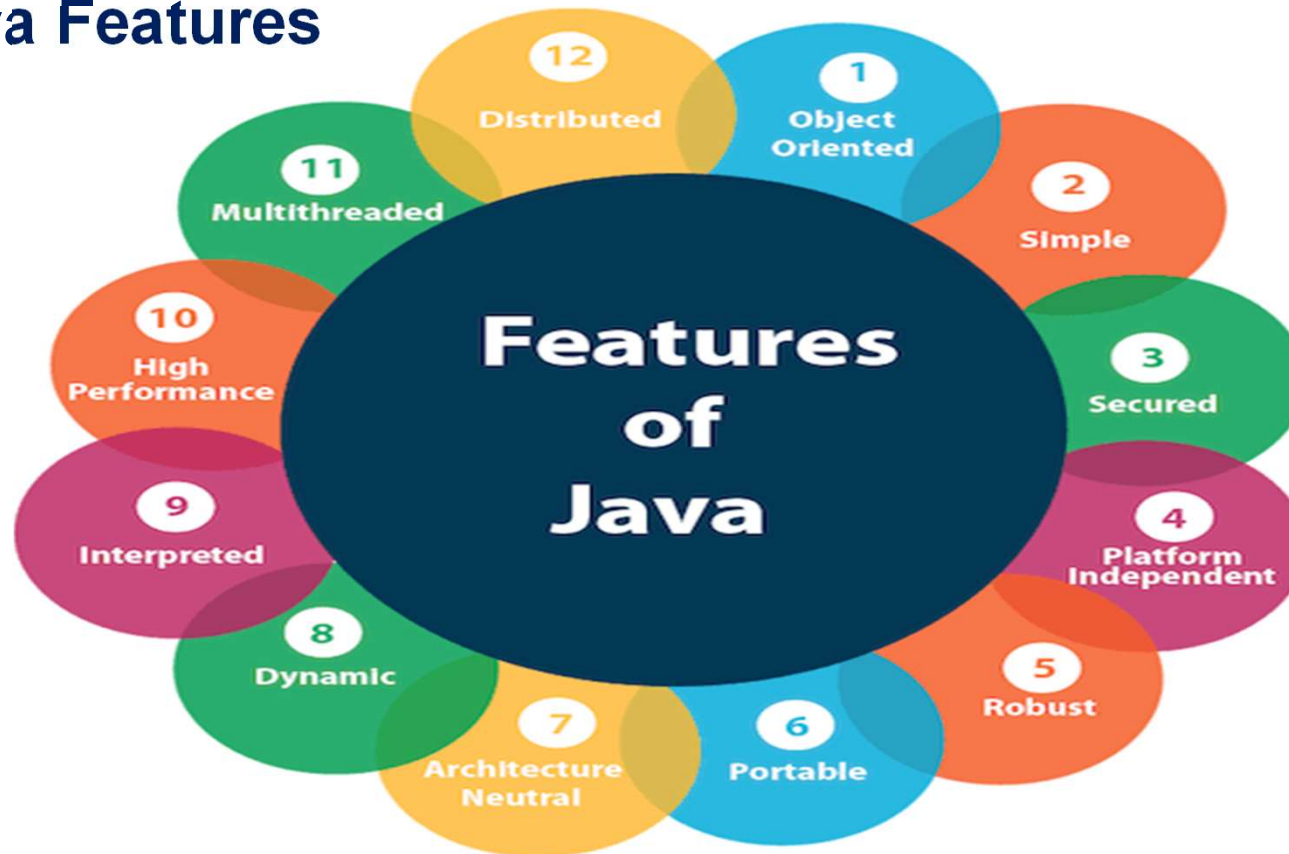
- JAVA was developed by James Gosling at Sun Microsystems_Inc in the year 1991.
- Java makes writing, compiling, and debugging programming easy.
- It helps to create reusable code and modular programs
- Java is a class-based, object-oriented programming language



<https://images.app.goo.gl/ESM8tY4uRyoFoYxv9>



Java Features



<https://static.javatpoint.com/images/core/java-features.png>



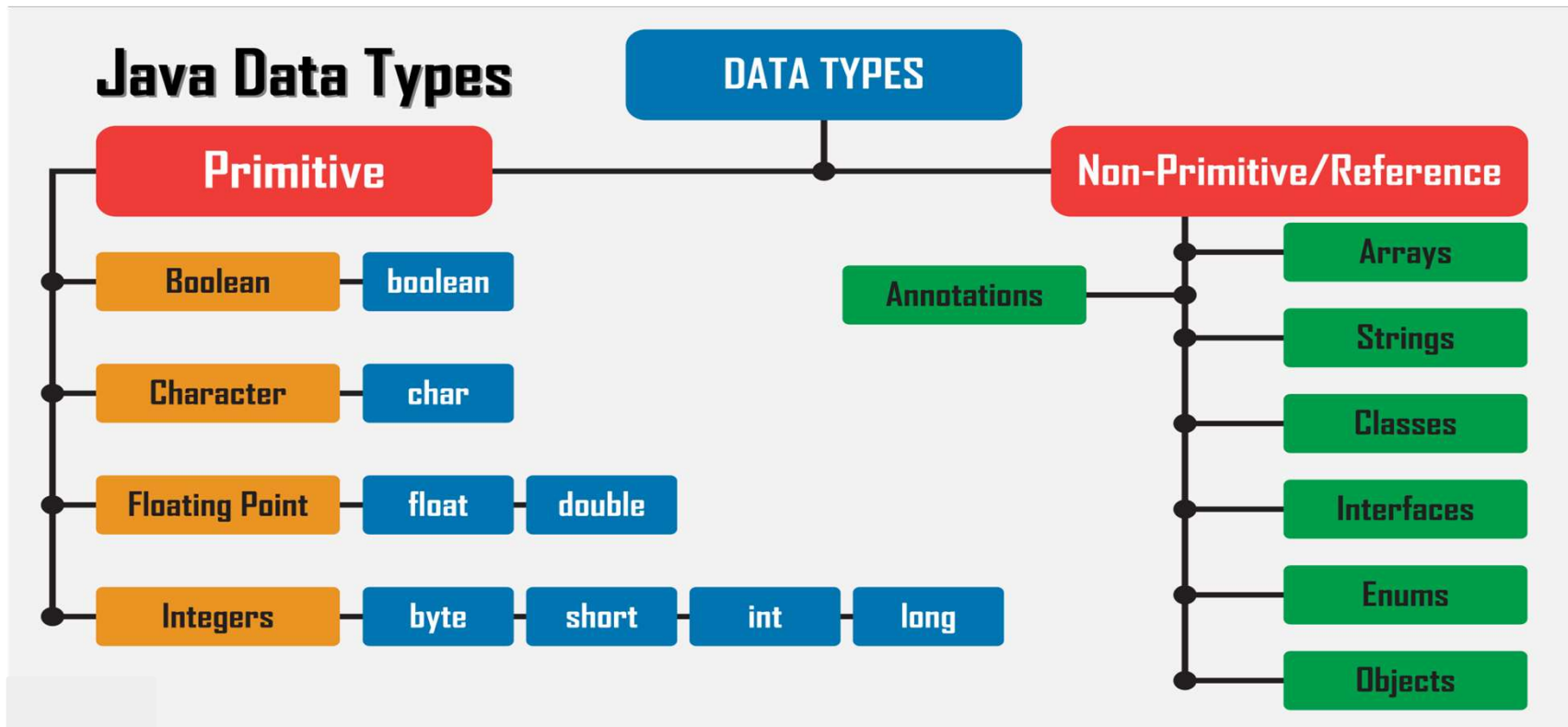
OOPS Concepts



<https://usupdates.com/wp-content/uploads/2020/11/OOPs-Concepts-in-Java.png>

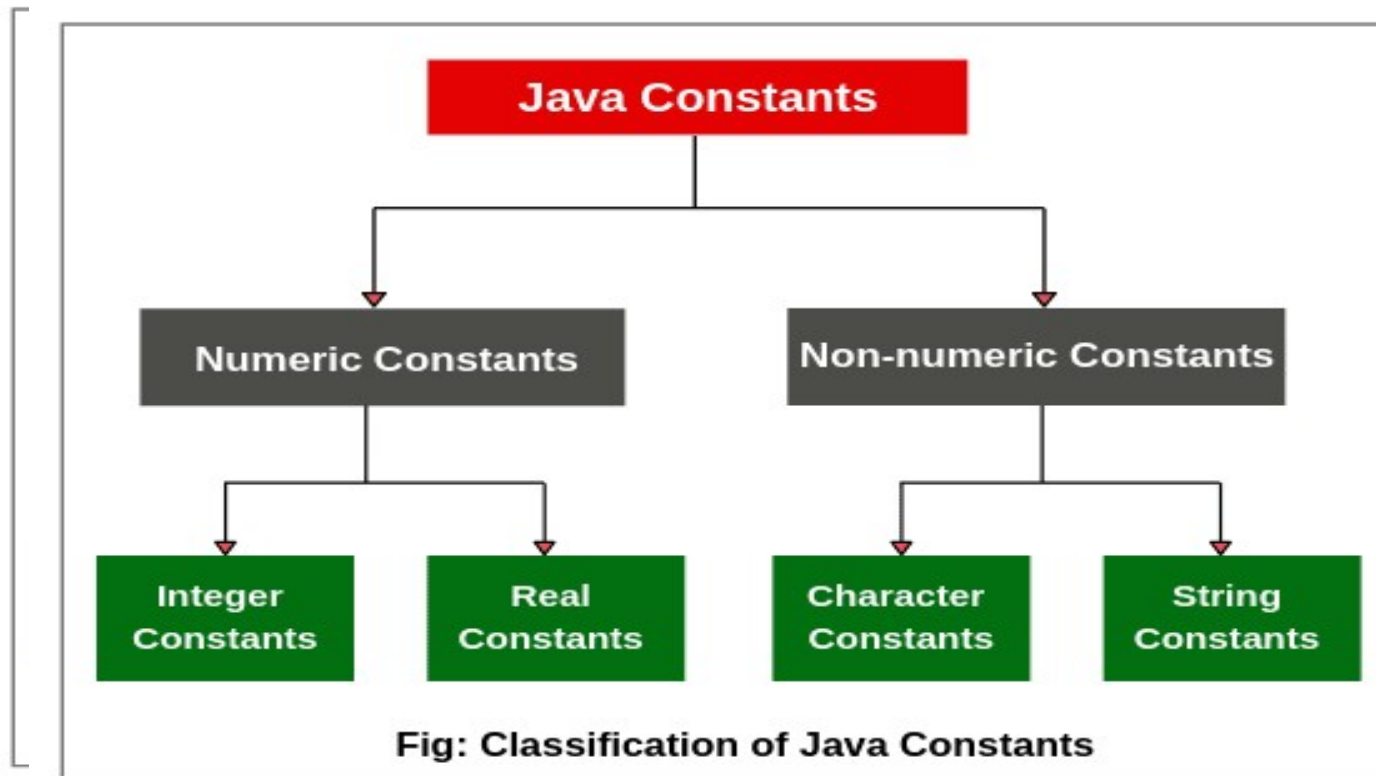


Data Types in Java



<https://getkt.com/wp-content/uploads/2019/12/Complete-set-of-Java-DATA-TYPES.png>

Constants in Java



<http://www.atnyla.com/library/images-tutorials/constants-in-java-1.PNG>



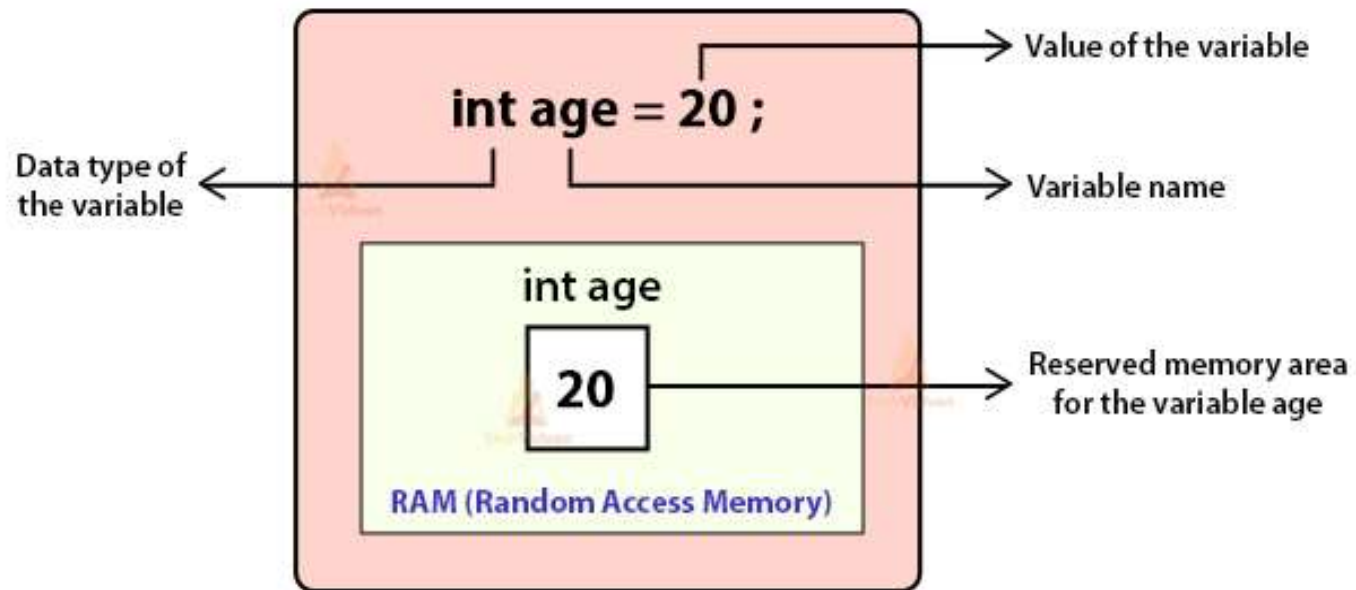
How to declare constant in Java?

- To define a variable as a constant, we just need to add the keyword “**final**” in front of the variable declaration.
- `final float pi = 3.14f;`



Variables in Java

Java Variable Declaration & its Memory Allocation



Operator in Java



<https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/01/Operators-in-Java-DF.jpg>



Operator Precedence

- Evaluate $a + b * c$

- multiplication first?
- addition first?

$a + (b * c)$

$(a + b) * c$

- Java solves this problem by assigning priorities to operators (**operator precedence**)

- operators with high priority are evaluated **before** operators with low priority
- operators with equal priority are evaluated **left to right**

Operator priority
(highest to lowest)

1. ()
2. * / %
3. + -
4. =

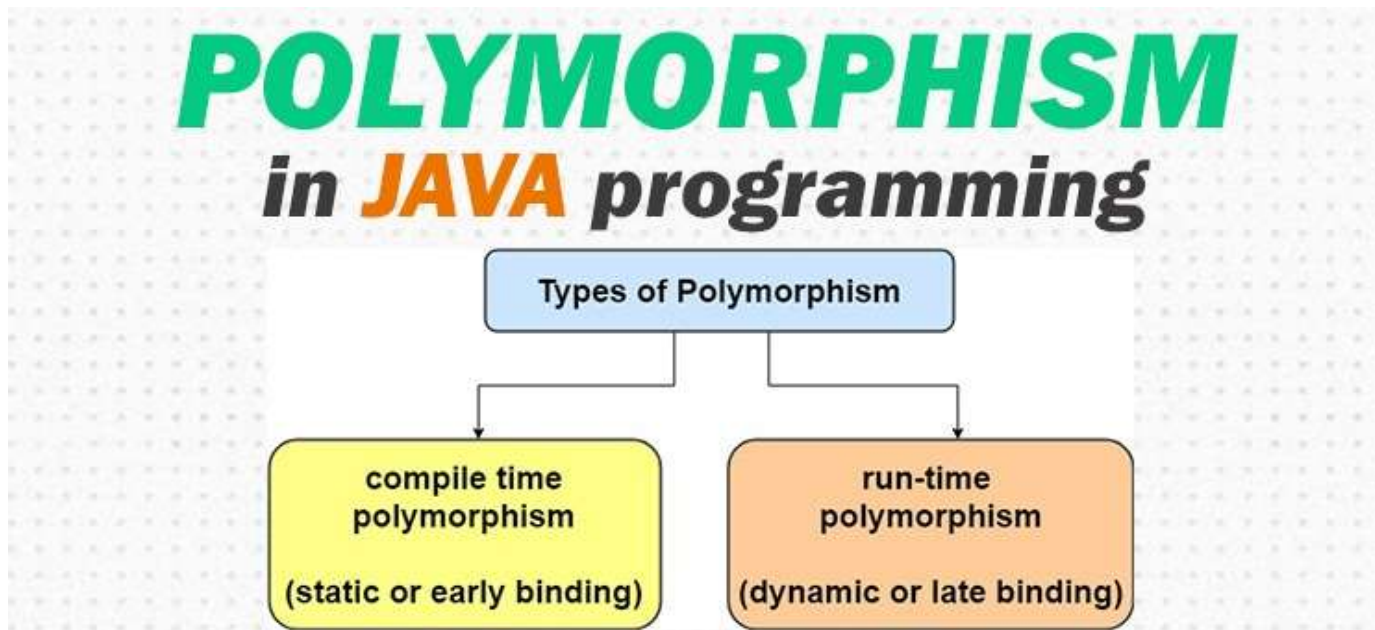
Types of Decision Making in Java



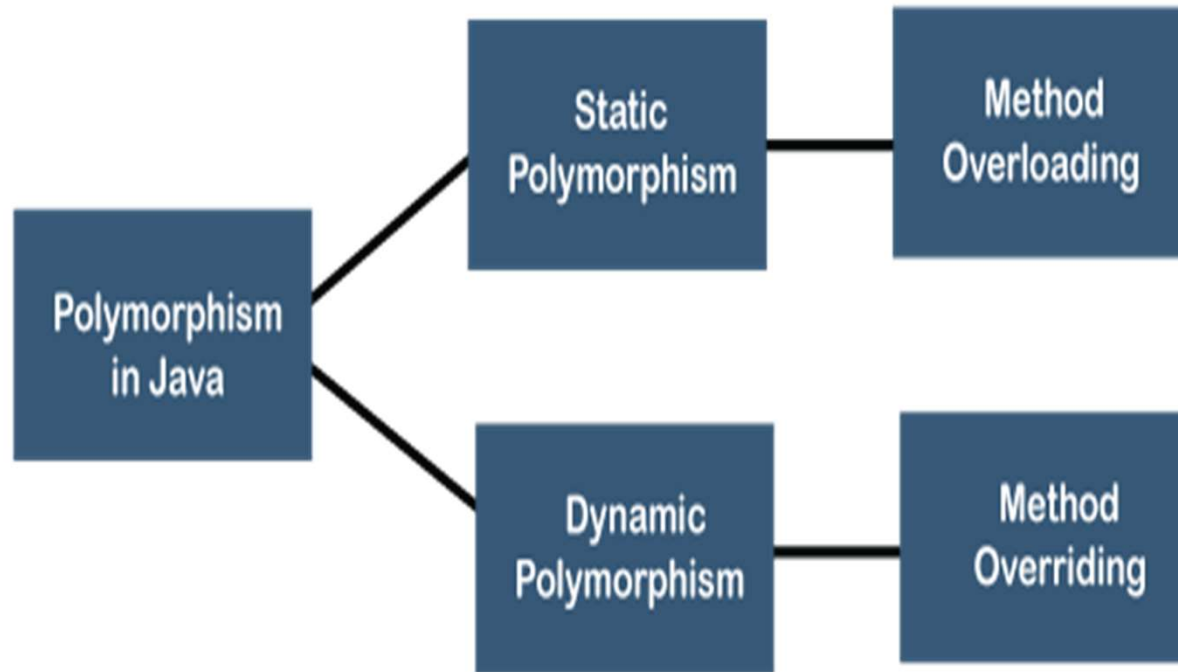
<https://i0.wp.com/techvidvan.com/tutorials/wp-content/uploads/sites/2/2020/02/types-of-decision-making-in-java-2.jpg?fit=802%2C420&ssl=1>

Polymorphism in Java

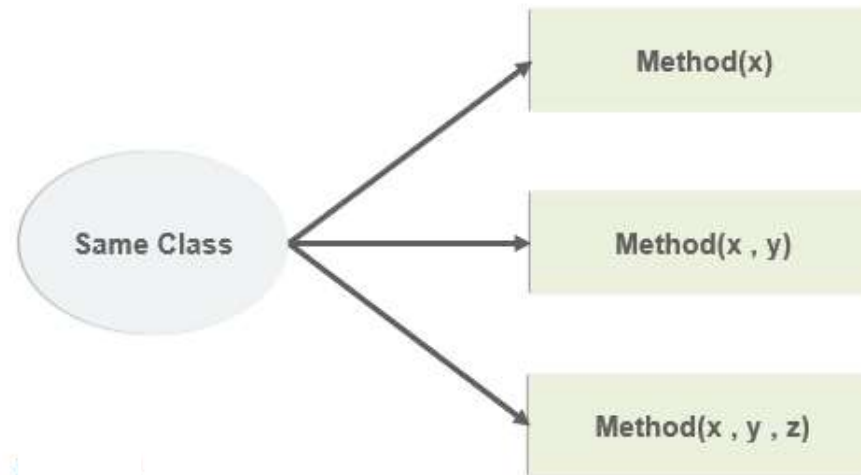
- **Polymorphism in Java** is a concept by which we can perform a *single action in different ways*.



<https://simplesnippets.tech/wp-content/uploads/2018/05/java-polymorphism-and-types.jpg>

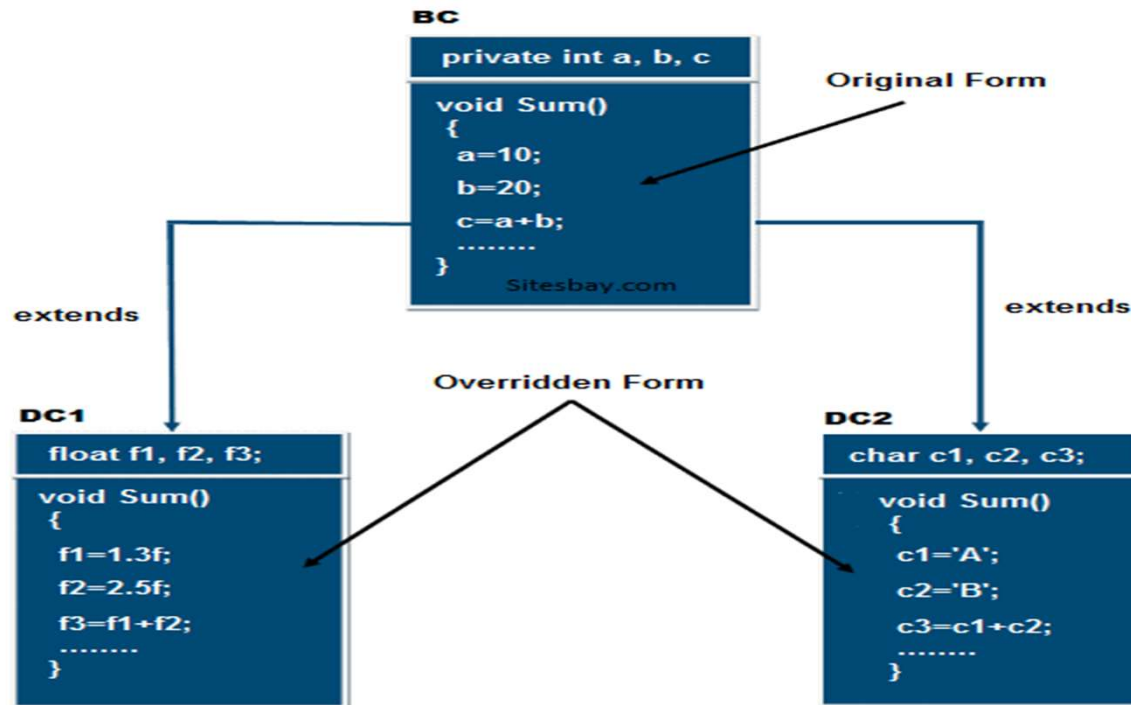


Example for Static Polymorphism



<https://www.edureka.co/blog/wp-content/uploads/2019/07/mov.png>

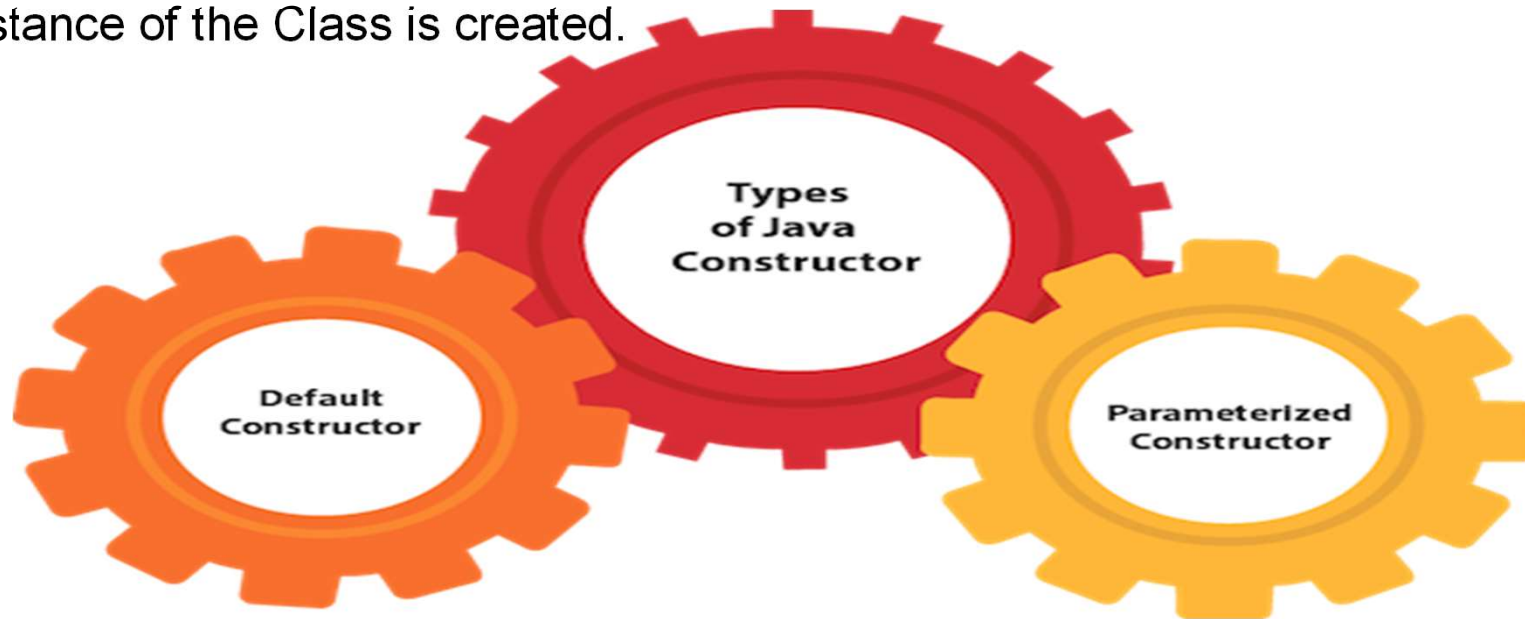
Example for Dynamic Polymorphism



<https://www.sitesbay.com/java/images/polymorphism-in-java.png>

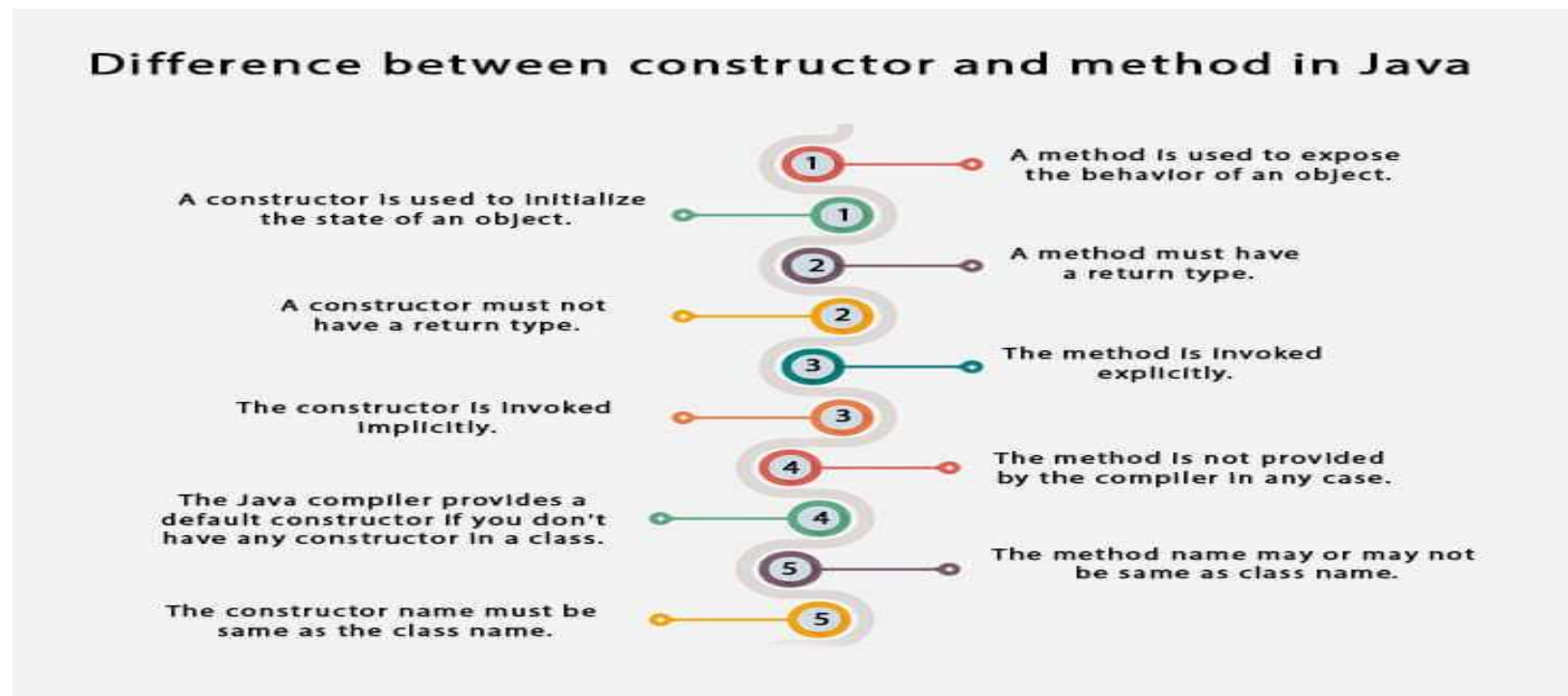
Constructors in Java

- A constructor is a block of codes similar to the method. It is called when an instance of the Class is created.



<https://images.app.goo.gl/JMfi3Za33Dtq6pUa8>

Constructors & Method



<https://static.javatpoint.com/images/constructor-vs-method-in-java.jpg>



Java Arrays

- **Array** is an object which contains elements of a similar data type.
- Types
 1. Single Dimensional Array
 2. Multi Dimensional Array



Strings in Java

- String is a sequence of characters. But in Java, string is an object that represents a sequence of characters.
- The `java.lang.String` class is used to create a string object
- There are two ways to create String object:
 1. By string literal
 2. By new keyword



Creating Strings in Java

1. String Literal

Java String literal is created by using double quotes. For Example:

```
String s="welcome";
```

2. By new keyword

```
String s=new String("Welcome");
```

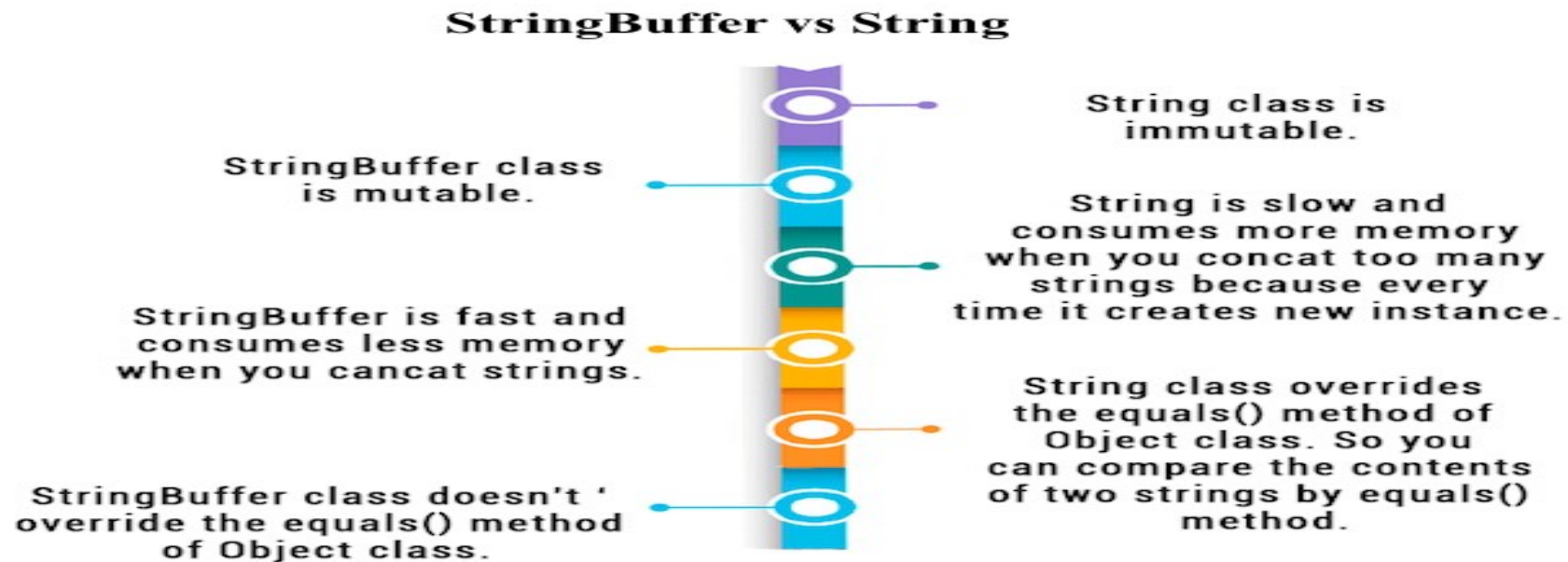


String Manipulation



<http://d1jnx9ba8s6j9r.cloudfront.net/blog/wp-content/uploads/2017/05/Java-String.png>

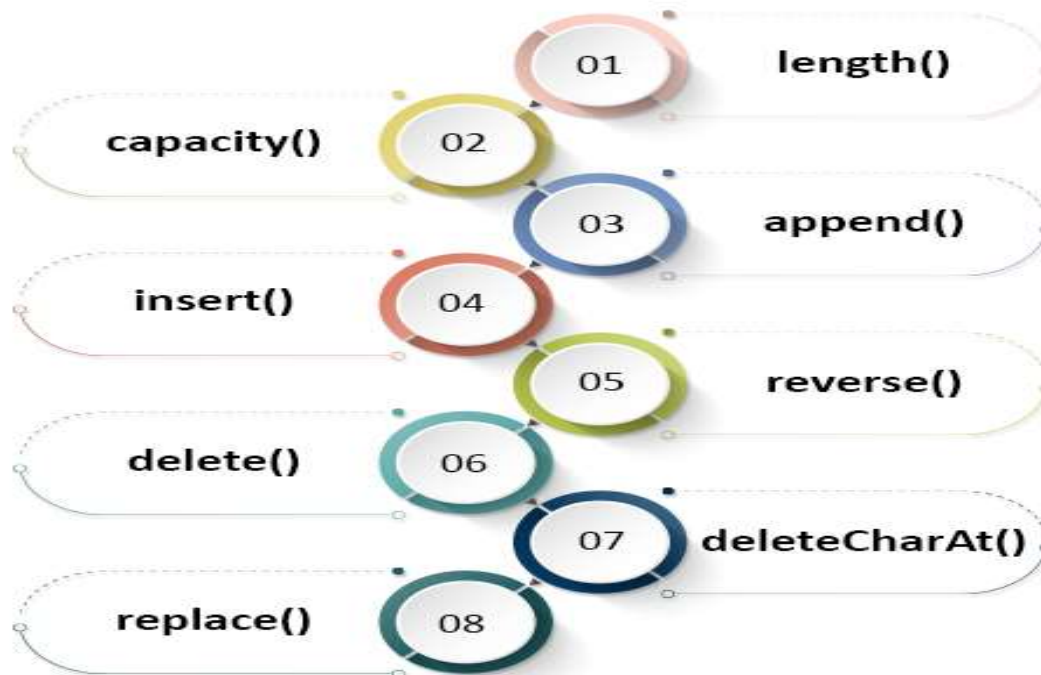
Difference between String and String Buffer class



<https://static.javatpoint.com/images/string-vs-stringbuffer.png>

String Buffer Methods

StringBuffer Methods in Java



<https://techvidvan.com/tutorials/wp-content/uploads/sites/2/2020/06/StringBuffer-Methods-in-Java.jpg>



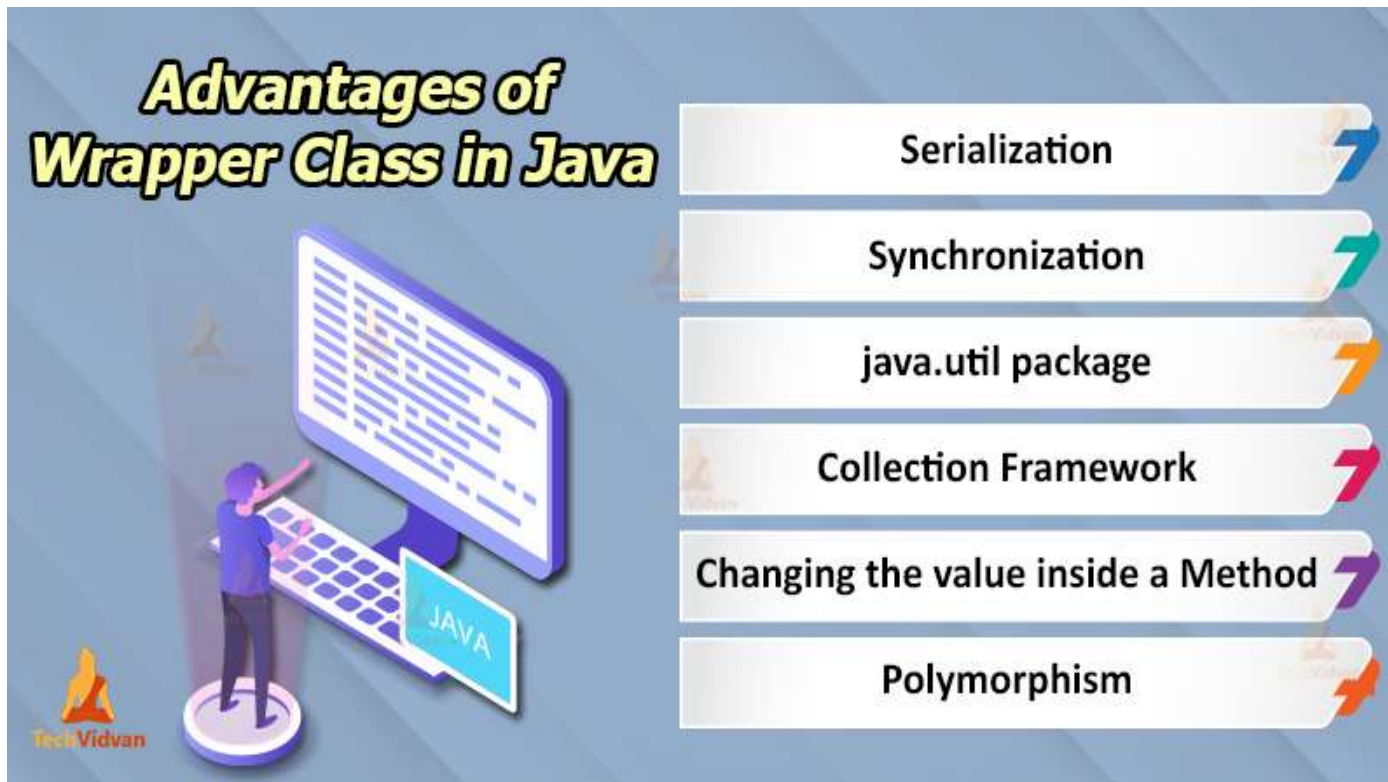
Wrapper classes in Java

- The wrapper class in Java provides the mechanism *to convert primitive into object and object into primitive.*

Wrapper Classes

Primitive Data Type	Wrapper Class
<i>double</i>	<i>Double</i>
<i>float</i>	<i>Float</i>
<i>long</i>	<i>Long</i>
<i>int</i>	<i>Integer</i>
<i>short</i>	<i>Short</i>
<i>byte</i>	<i>Byte</i>
<i>char</i>	<i>Character</i>
<i>boolean</i>	<i>Boolean</i>

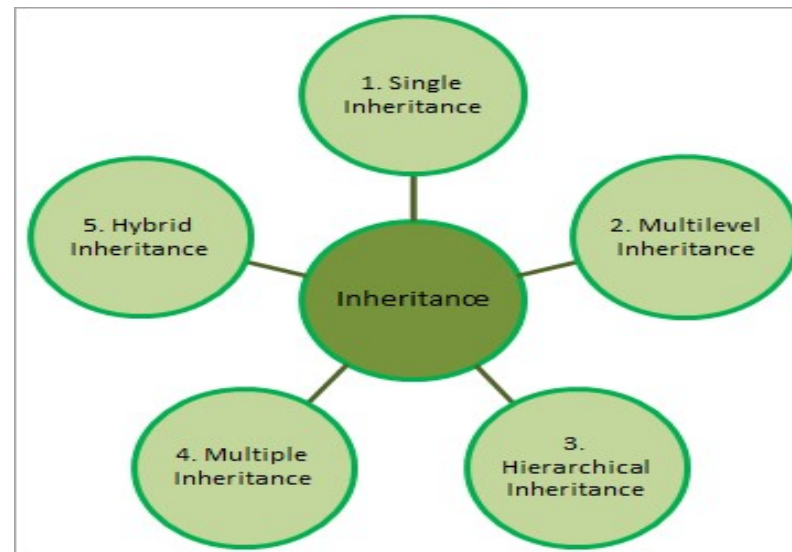
Advantages of Wrapper Class



<https://techvidvan.com/tutorials/wp-content/uploads/sites/2/2020/03/advantages-of-wrapper-class-in-java.jpg>

Inheritance in Java

- Its a mechanism in which one object acquires all the properties and behaviors of a parent object.



<https://encryptedtbn0.gstatic.com/images?q=tbn:ANd9GcSiVtoOtZRt9R8whZs2OngrxoM0UHU5MzI6bA&usqp=CAU>

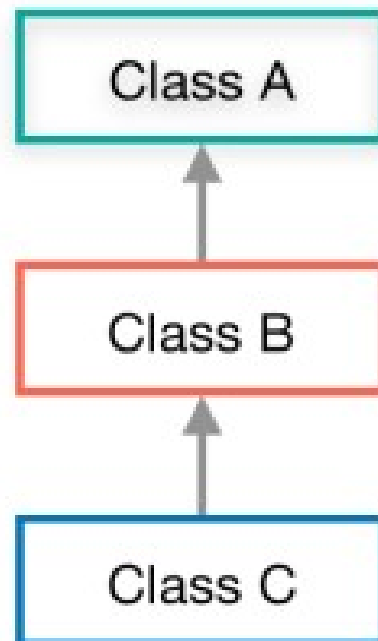
Single Inheritance





Multilevel Inheritance

Multilevel
Inheritance



```
public class A {  
    .....  
}  
  
public class B extends A {  
    .....  
}  
  
public class C extends B {  
    .....  
}
```

Interface in Java

- It is a blueprint of a class. It has static constants and abstract methods.



<https://static.javatpoint.com/interview/images/why-use-java-interface.jpg>

Interface Example

```

public interface Human
{
    int a=20;
    public abstract void walk();
    public abstract void eat();
}
class Engineer implements Human
{
    @Override
    public void eat()
    {
        System.out.println("Engineer can eat ");
    }
    @Override
    public void walk()
    {
        System.out.println("Engineer can walks");
    }
    public static void main(String[] args)
    {
        Engineer E=new Engineer();
        E.eat();
        E.walk();
    }
}

```

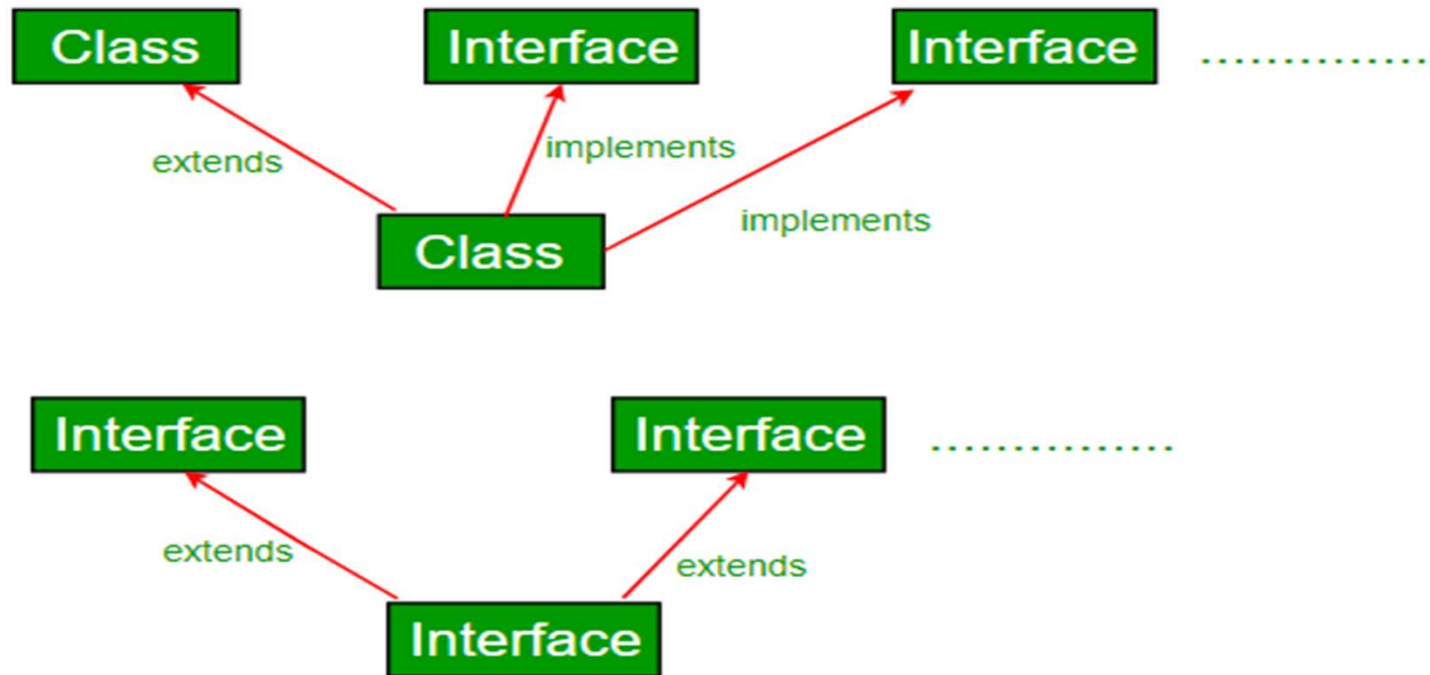
OUTPUT:
Engineer can eat
Engineer can walks

Diagram:

- Human** is labeled as an **interface**.
- Engineer** is labeled as a **class**.
- A red dashed arrow labeled **implements** points from the **Engineer** class to the **Human** interface.
- White arrows point from the text **abstract method declaration** to the `public abstract void walk();` and `public abstract void eat();` lines in the code.
- White arrows point from the text **Since Engineer class implements Human interface overriding of abstract methods is mandatory** to the `@Override` annotations above the `eat()` and `walk()` methods in the **Engineer** class.

<https://bytesofgigabytes.com/IMAGES/java/Interface/InterfaceAccess.png>

Relationship between Class and Interface



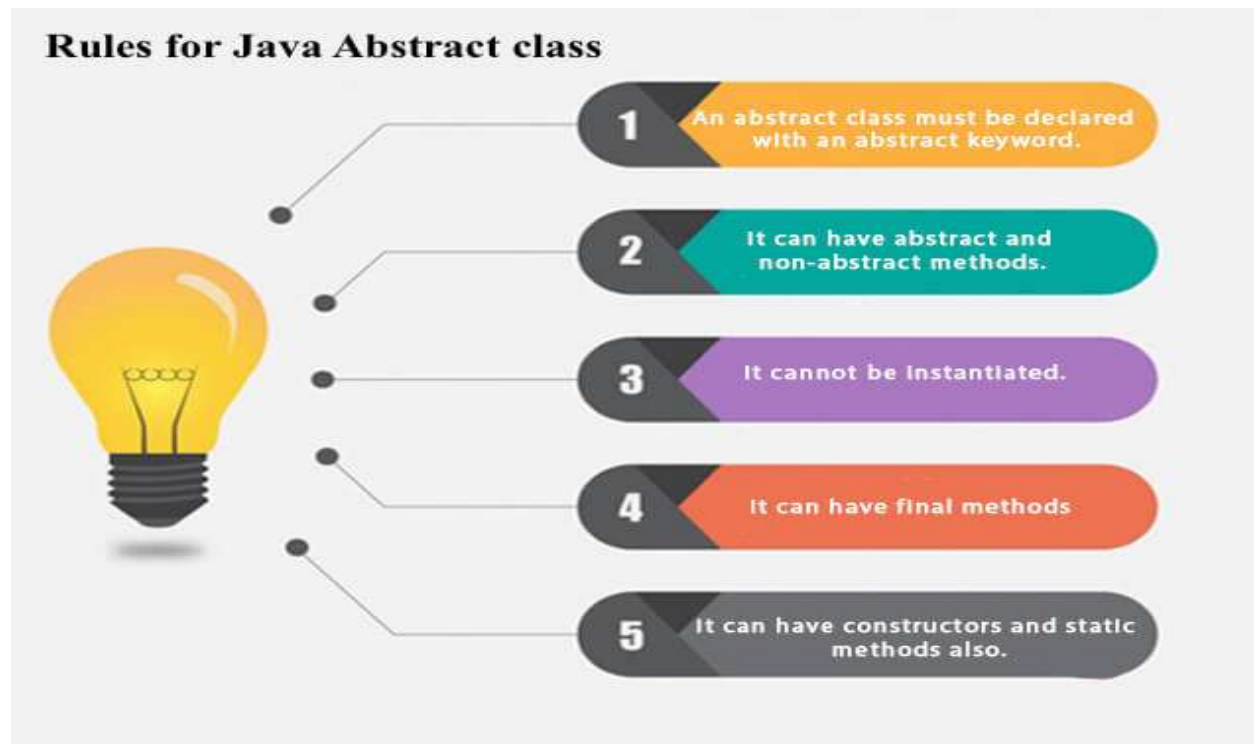
<https://media.geeksforgeeks.org/wp-content/cdn-uploads/extends.png>



Abstract Class

- A class which is declared with the abstract keyword is known as an abstract class. It can have abstract and non-abstract methods.
- Abstraction is a process of hiding implementation details and showing only functionality to the user.
- There are two ways to achieve abstraction in java
 1. Abstract class
 2. Interface

Rules for Abstract Class



<https://static.javatpoint.com/images/abstract-class-in-java.jpg>

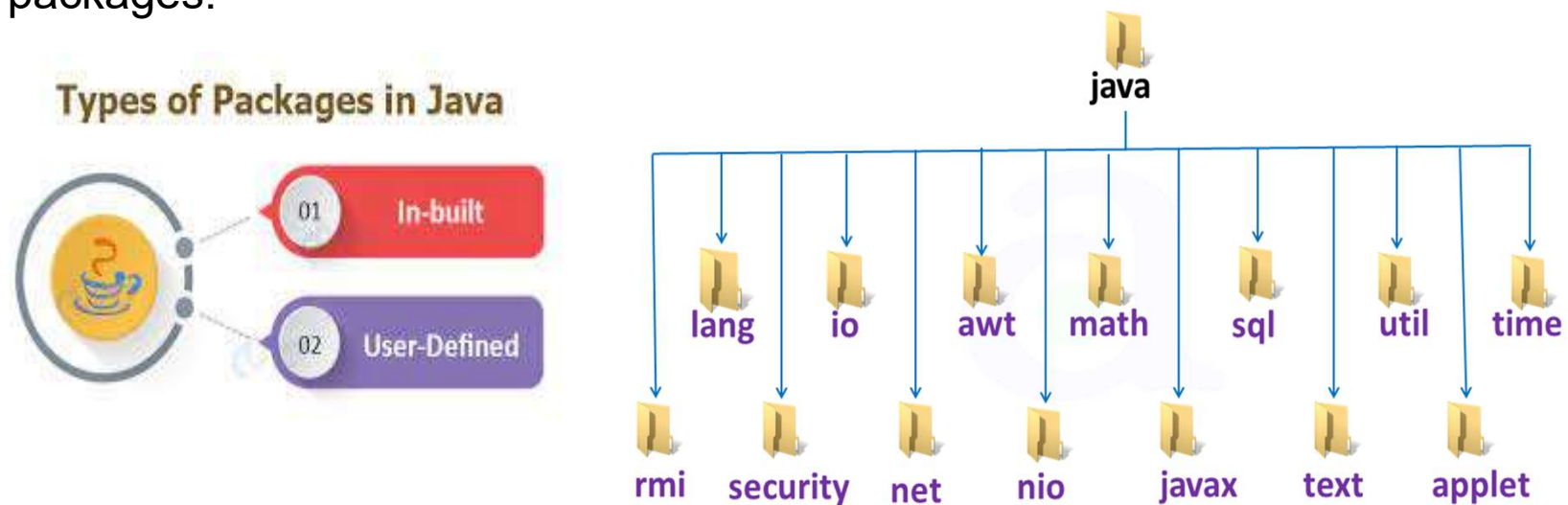
Difference between Abstract class and Interface

Difference between Abstract Classes & Interfaces		
Abstract Classes		Interfaces
Abstract classes are fast.	Speed	Interfaces are slow.
Abstract classes can extend only one class.	Multiple Inheritance	Interface can implement several interfaces.
You can define fields as well as constants.	Defined Fields	You cannot define fields in an interface.
A single abstract class can extend one & only one interface.	Extension Limit	A single interface can extend multiple interfaces.

<https://data-flair.training/blogs/difference-between-abstract-class-and-interface-in-java/>

Packages in Java

- A java package is a group of similar types of classes, interfaces and sub-packages.



<http://www.atnyla.com/library/images-tutorials/built-in-packages-in-java.PNG>

Advantages of Packages

Resolve naming conflict



Reuse of code



Organization of project



Modularity



Access protection



Information hiding



Access Specifiers in Java

Access Specifiers in Java

		public	private	protected	default
Same Package	Class	YES	YES	YES	YES
	Sub class	YES	NO	YES	YES
	Non sub class	YES	NO	YES	YES
Different Package	Sub class	YES	NO	YES	NO
	Non sub class	YES	NO	NO	NO

<https://usemynotes.com/wp-content/uploads/2021/02/what-are-access-specifiers-in-java.jpg>



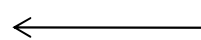
Inner Classes in Java

- Java inner class or nested class is a class which is declared inside the class or interface.
- We use inner classes to logically group classes and interfaces in one place so that it can be more readable and maintainable.

```
class Java_Outer_class{
```

```
    //code
```

```
        class Java_Inner_class{
```



Inner Class Syntax

```
            //code
```

```
        }
```

```
    }
```



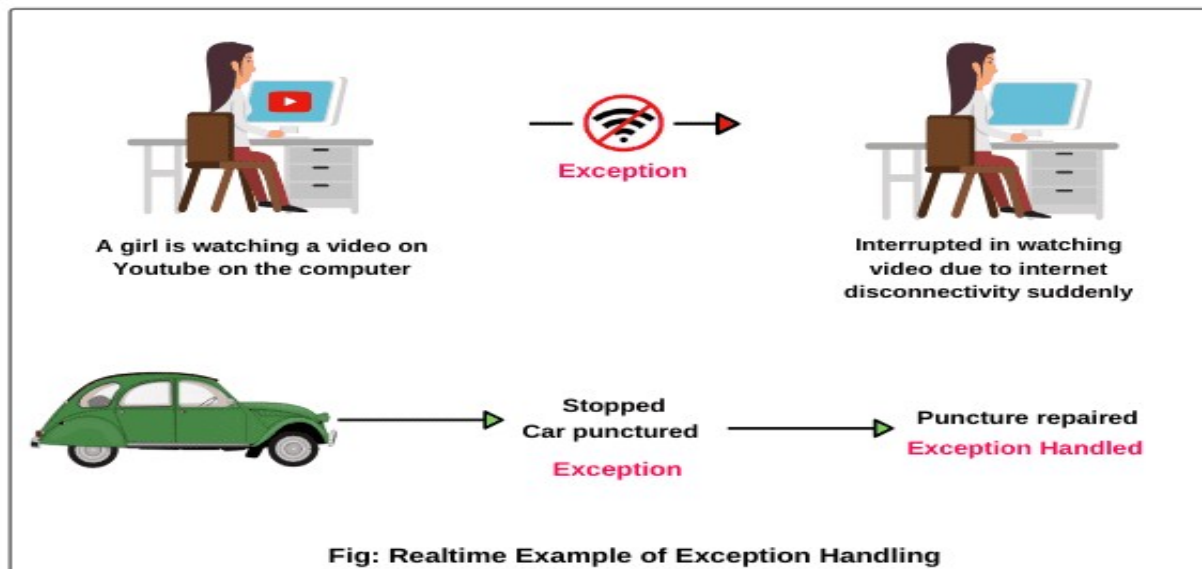
Types of Inner Class



<https://techvidvan.com/tutorials/wp-content/uploads/sites/2/2020/02/types-of-java-inner-class.jpg>

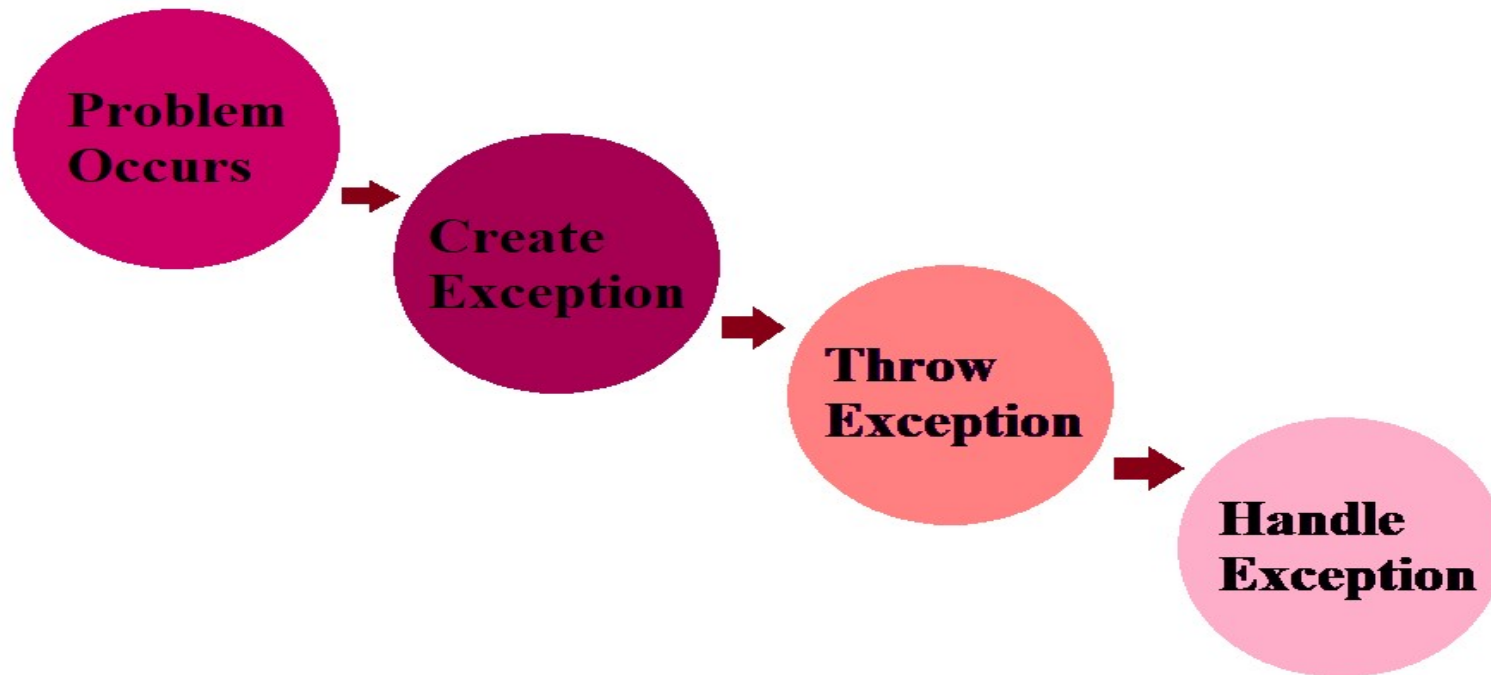
Exception Handling in Java

- In Java, an exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.



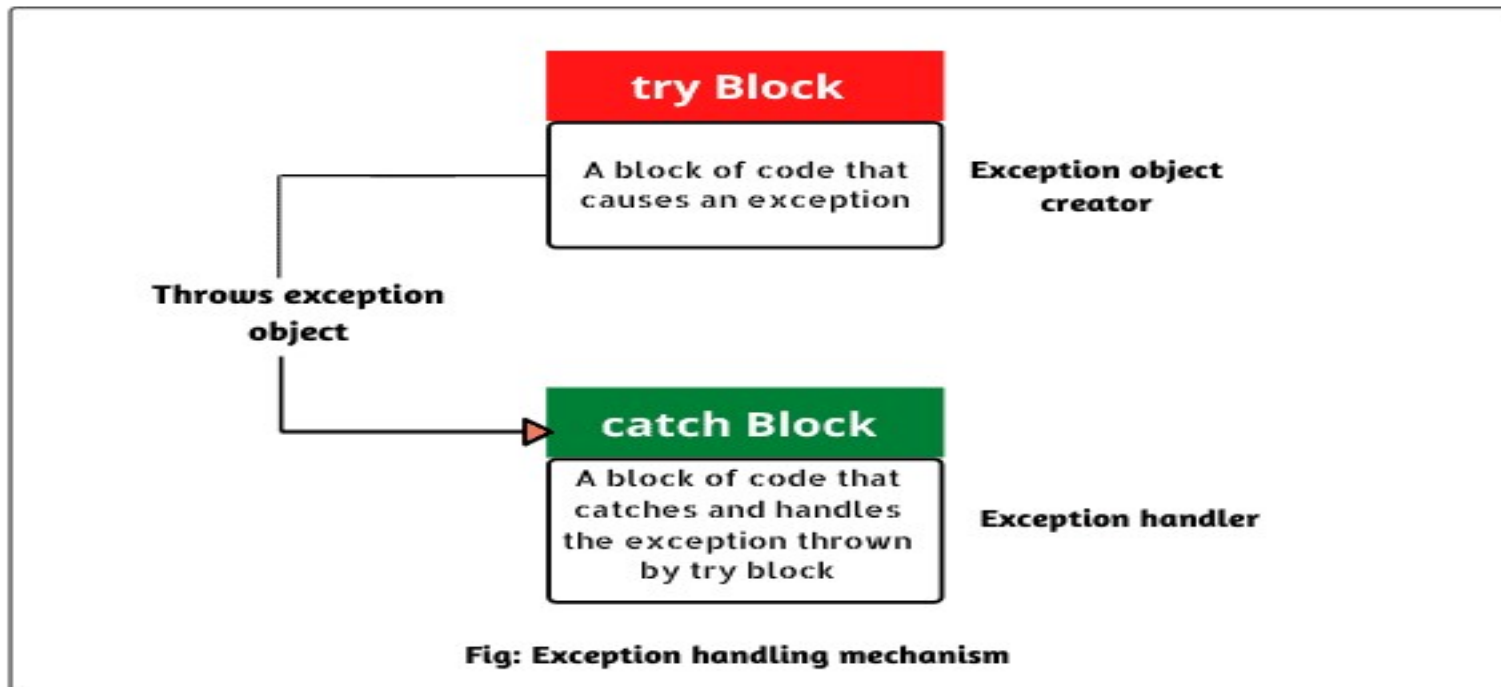
<https://www.scientecheasy.com/wp-content/uploads/2020/02/exception-handling.png>

Exception Handling Mechanism



<https://images.app.goo.gl/vDnN7k4MrDDwN6Pc7>

Try-Catch Block



<https://www.scientecheasy.com/wp-content/uploads/2020/02/try-catch-block.png>

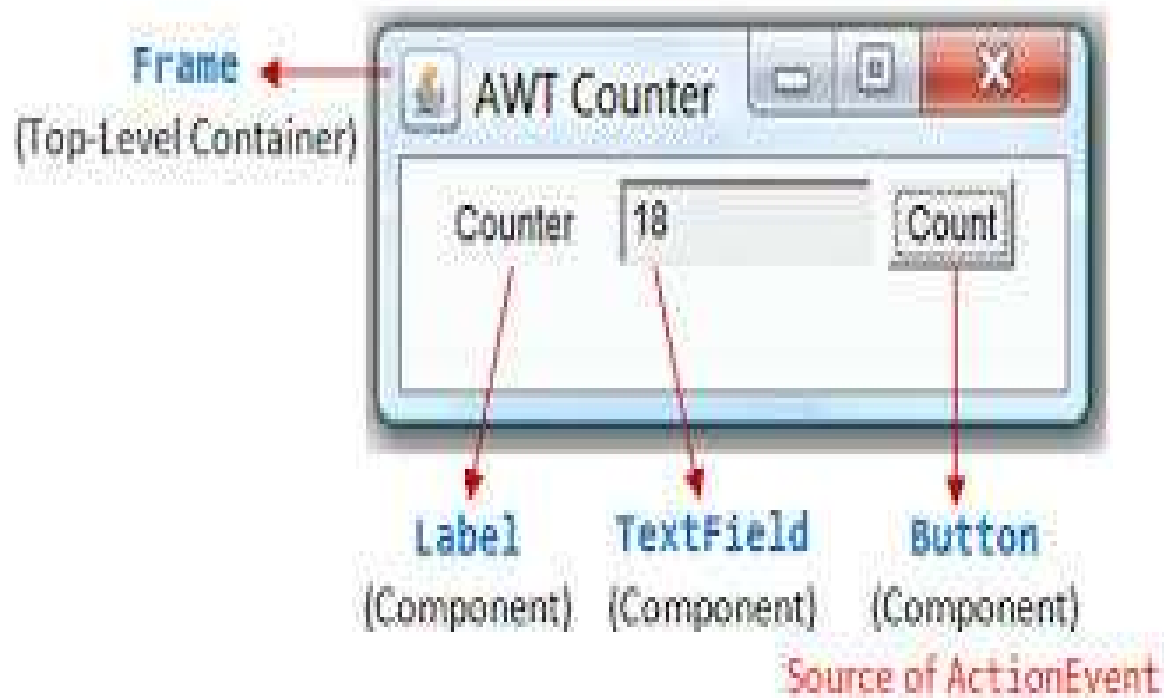


Java AWT

- **AWT** (Abstract Window Toolkit) is *an API to develop GUI or window-based applications* in java.
- AWT components are platform-dependent.
- AWT is heavyweight
- The java.awt package provides classes for AWT API.

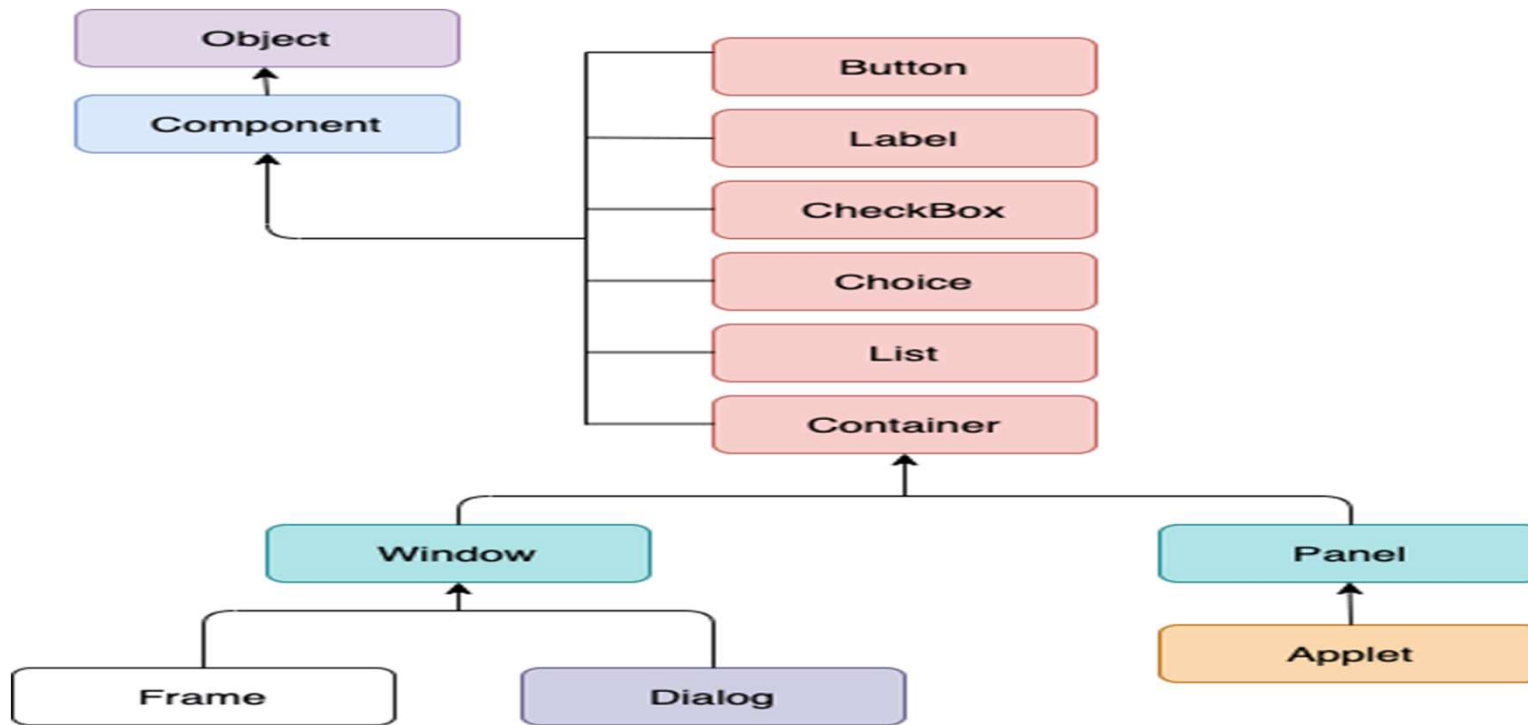


Java AWT





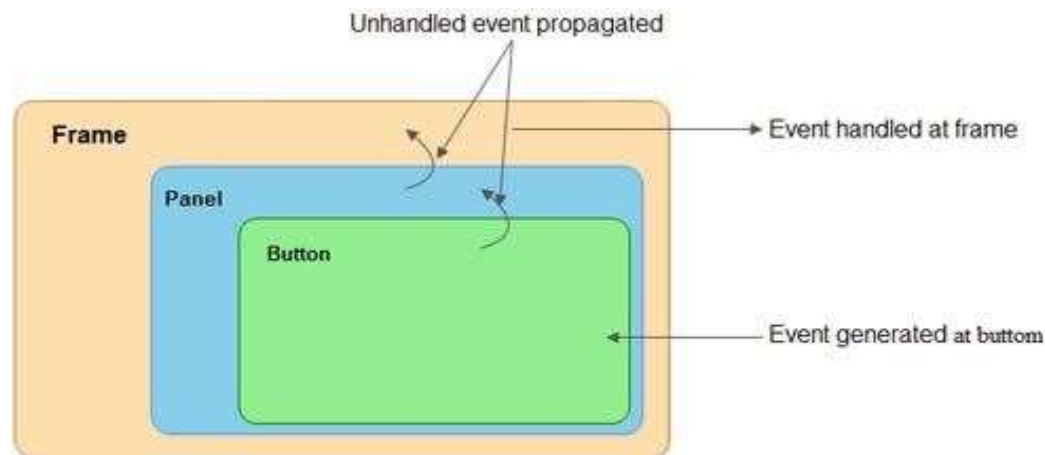
Java AWT Hierarchy



<https://static.javatpoint.com/core/images/awt-program-in-java.png>

Event Handling in Java

- Changing the state of an object is known as an event.
- For example, click on button, dragging mouse etc. The `java.awt.event` package provides many event classes and Listener interfaces for event handling.





Some Events and Listeners in Java

EVENTS	SOURCE	LISTENERS
Action Event	Button, List,MenuItem, Text field	ActionListener
Component Event	Component	Component Listener
Focus Event	Component	FocusListener
Item Event	Checkbox,CheckboxMen uitem, Choice, List	ItemListener
Key Event	when input is received from keyboard	KeyListener
Text Event	Text Component	TextListener
Window Event	Window	WindowListener
Mouse Event	Mouse related event	MouseListener

<https://images.app.goo.gl/t3MoNBSEAMTmQwBU6>



Develop Web pages using Java.
(30 Hours)



In this section, we will discuss:

- What is servlets, Servlet package.
- Setting up servlet environment, servlet life cycle.
- Servlets form data.
- Servlet client HTTP request, HTTP server response.
- Status codes.
- Filters.
- Exception Handling.
- Cookies & Sessions.
- Database connectivity.
- Servlets date & time.
- Auto page refresh.

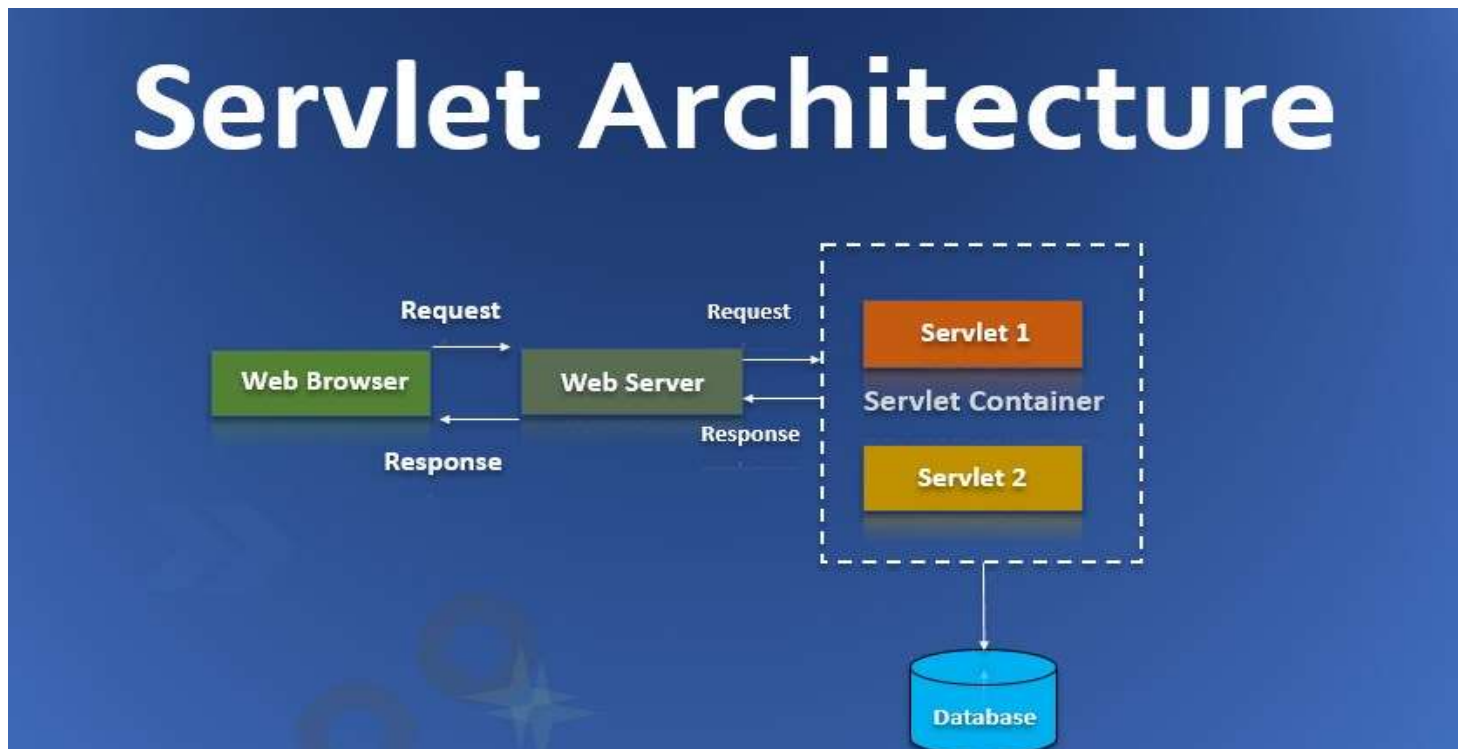


What is Servlet?

- It used to create a web application (resides at server side and generates a dynamic web page).
- Servlet is an API that provides many interfaces and classes including documentation.
- Servlet is a web component that is deployed on the server to create a dynamic web page.



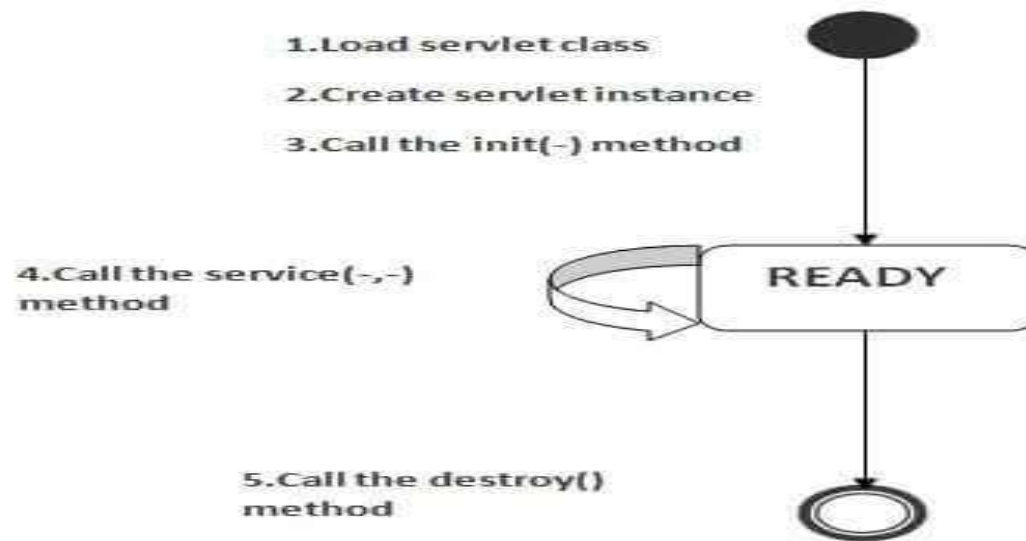
Servlet Architecture



<https://cdn.educba.com/academy/wp-content/uploads/2020/01/Servlet-Architecture-Main.jpg>



Life Cycle of a Servlet



<https://static.javatpoint.com/images/servletlife.jpg>



Reading Form Data using Servlet

- Servlets handles form data parsing automatically using the following methods.
- **getParameter()** – You call `request.getParameter()` method to get the value of a form parameter.
- **getParameterValues()** – Call this method if the parameter appears more than once and returns multiple values, for example checkbox.
- **getParameterNames()** – Call this method if you want a complete list of all parameters in the current request.

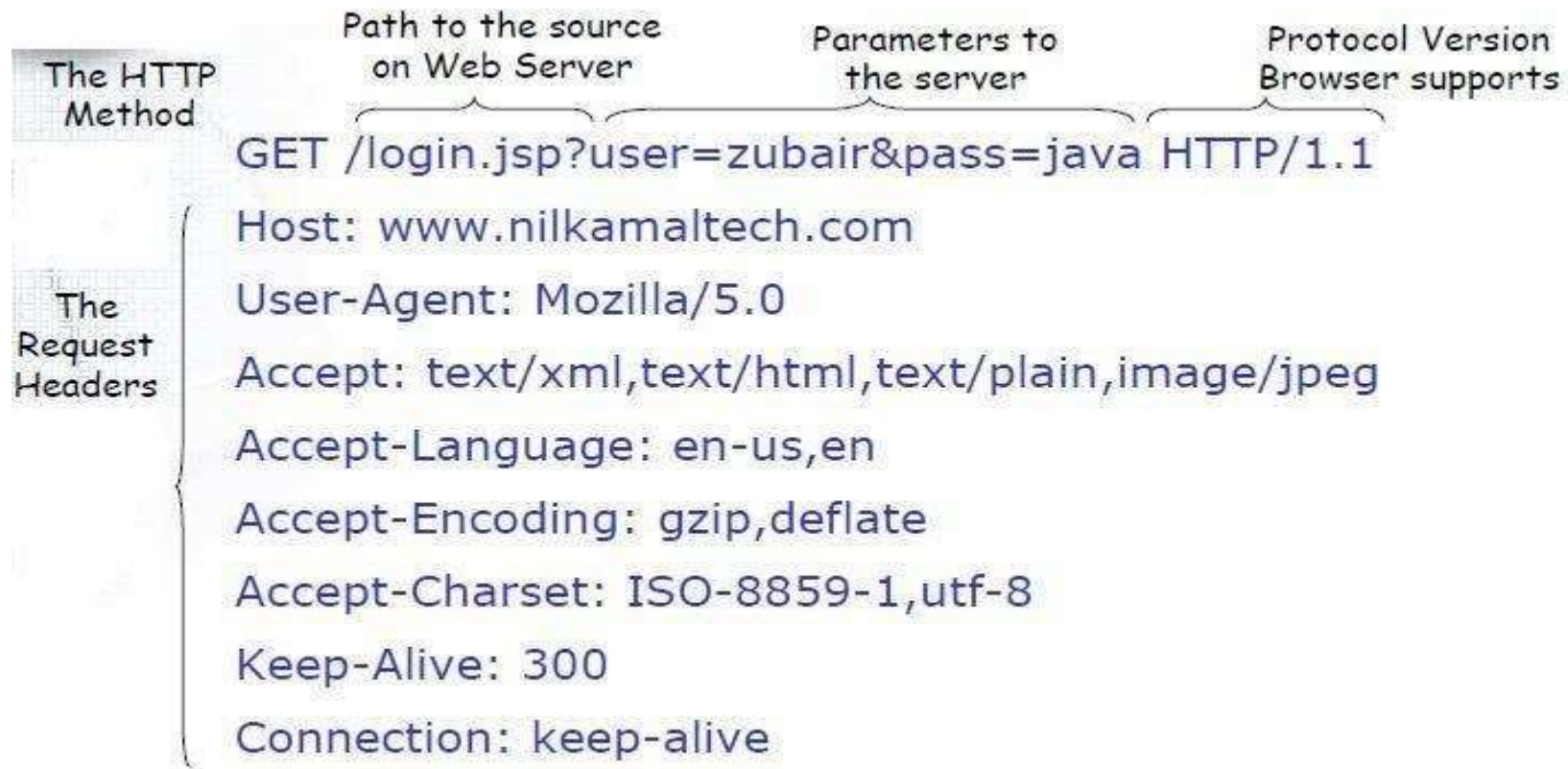
Servlet client HTTP Request



<https://static.javatpoint.com/servletpages/servletterminology/images/http-requests.jpg>

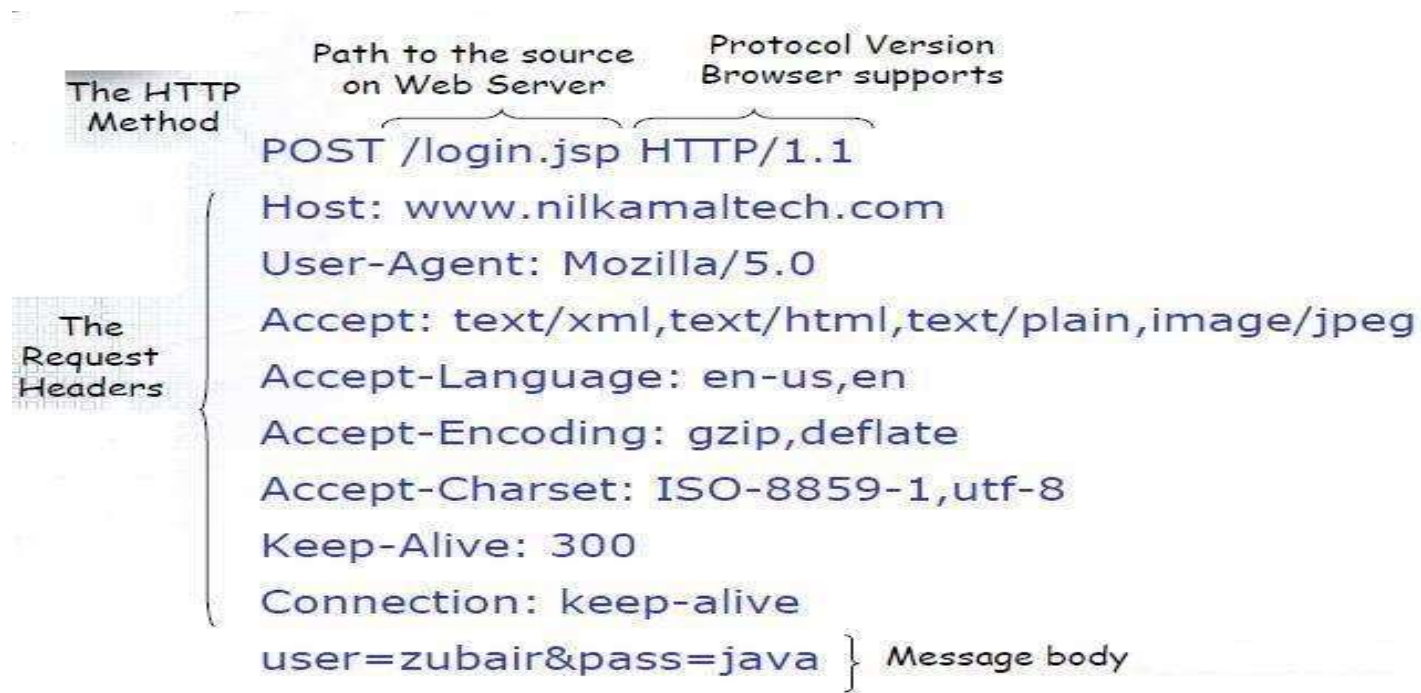


Anatomy of GET Request



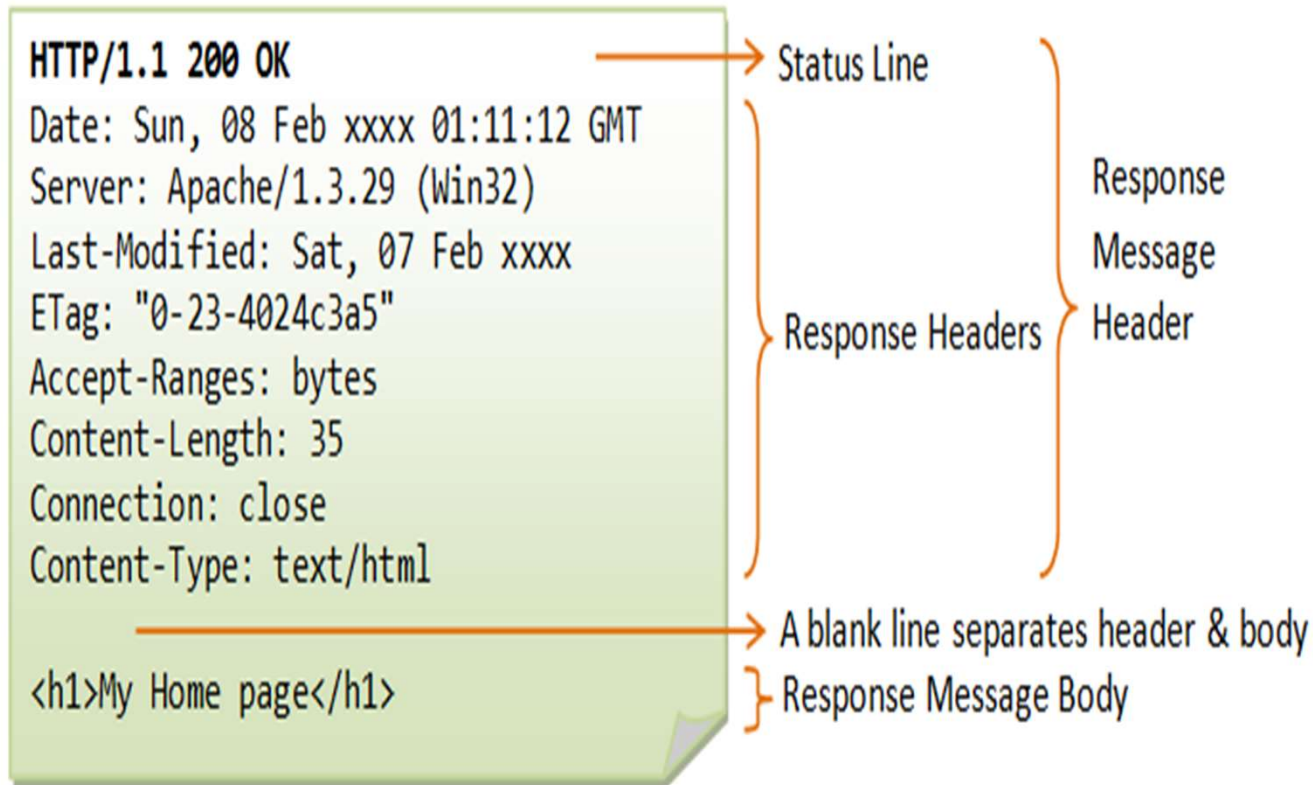


Anatomy of POST Request





HTTP server Response





Status Code

- The Server issues an HTTP Status Code in response to a request of the client made to the server.
- Status code is a 3-digit integer. The first digit of status code is used to specify one of five standard classes of responses.
- The last two digits of status code do not have any categorization role.

HTTP Status Codes



https://miro.medium.com/max/920/1*w_iicbG7L3xEQTArjHUS6g.jpeg



Servlet Filters

- A **filter** is an object that is invoked at the preprocessing and postprocessing of a request.
- It is mainly used to perform filtering tasks such as conversion, logging, compression, encryption and decryption, input validation etc.
- The **servlet filter is pluggable**, i.e. its entry is defined in the web.xml file



Usage of Filters

- Recording all incoming requests
- logs the IP addresses of the computers from which the requests originate
- conversion
- data compression
- encryption and decryption
- input validation etc.

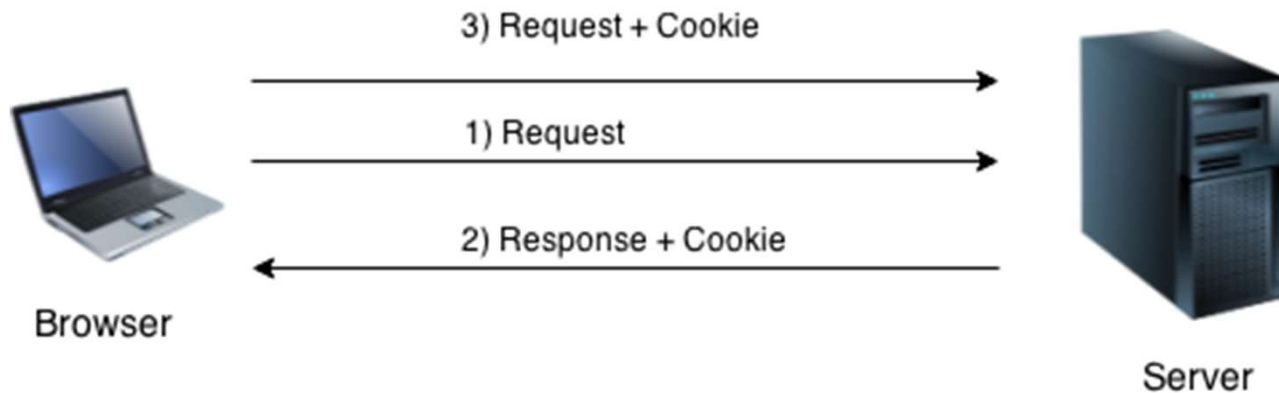


Advantage of Filter

- Filter is pluggable.
- One filter don't have dependency onto another resource.
- Less Maintenance

Cookies in Servlet

- A **cookie** is a small piece of information that is persisted between the multiple client requests.



<https://static.javatpoint.com/phppages/images/cookie.png>



Advantages & Disadvantages of Cookies

Advantages:

- Simplest technique of maintaining the state.
- Cookies are maintained at client side

Disadvantages:

- It will not work if cookie is disabled from the browser.
- Only textual information can be set in Cookie object.



Cookie Class

- `javax.servlet.http.Cookie` class provides the functionality of using cookies. It provides a lot of useful methods for cookies

Constructor	Description
<code>Cookie()</code>	constructs a cookie.
<code>Cookie(String name, String value)</code>	constructs a cookie with a specified name and value.



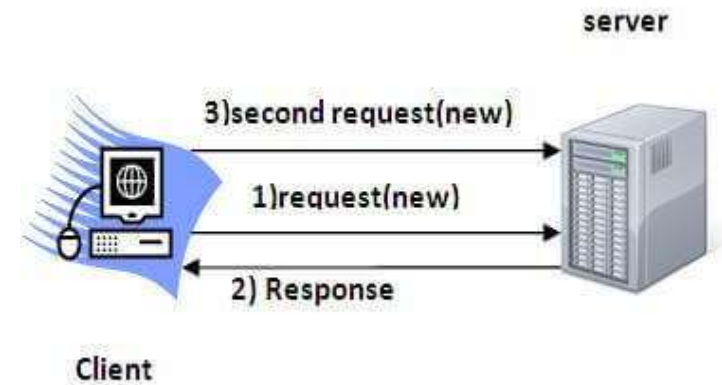
Sessions in Servlet

- **Session** simply means a particular interval of time.
- **Session Tracking** is a way to maintain state (data) of an user. It is also known as **session management** in servlet.
- HTTP is stateless that means each request is considered as the new request. It is shown in the figure given below:



Each time user requests to the server, server treats the request as the new request.

So we need to maintain the state of an user to recognize to particular user.



<https://static.javatpoint.com/images/newrequest.JPG>



Servlets Date and Time

- Java provides the Date class available in java.util package, this class encapsulates the current date and time.
- The Date class supports two constructors as shown in the following table.

Sr.No	Constructor & Description
1	Date() This constructor initializes the object with the current date and time.
2	Date(long millisec) This constructor accepts an argument that equals the number of milliseconds that have elapsed since midnight, January 1, 1970.



Getting Current Date and Time

```
import java.util.Date;

public class DateDemo {

    public static void main(String args[]) {

        // Instantiate a Date object

        Date date = new Date()

        // display time and date using toString()

        System.out.println(date.toString());

    }

}
```




Auto Page Refresh

- Java Servlet makes this job easy by providing you a mechanism where you can make a webpage in such a way that it would refresh automatically after a given interval.
- The simplest way of refreshing a web page is using method **setIntHeader()** of response object. Following is the signature of this method `public void setIntHeader(String header,int headerValue)`



edunet
foundation