Engineering in One Video (EIOV)

Watch video on YouTube

B.Tech

Web Technology

KCS-602

With Notes

UNIT-1

Introduction

(Web, Core Java, OOPs, Applet, AWT, Multithreading, Exception Handling, etc.)

(in one video)

AKTU Exam

Engineering in One Video (EIOV)

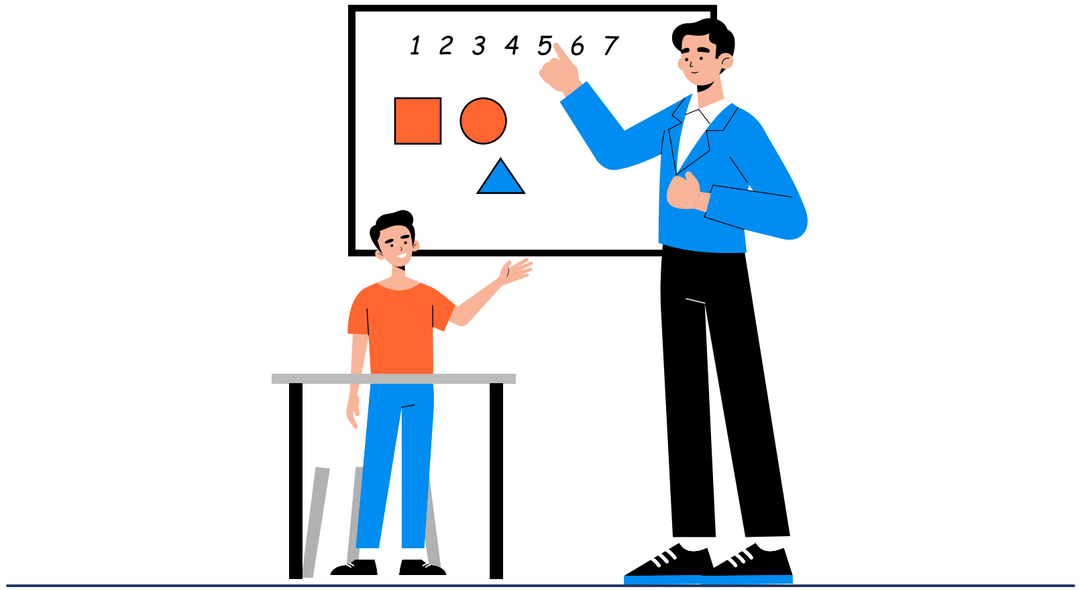
Watch video on YouTube

Topics to be covered...

History of Web
Protocol Governing Web
How to develop Web Project
Internet Services
Core Java

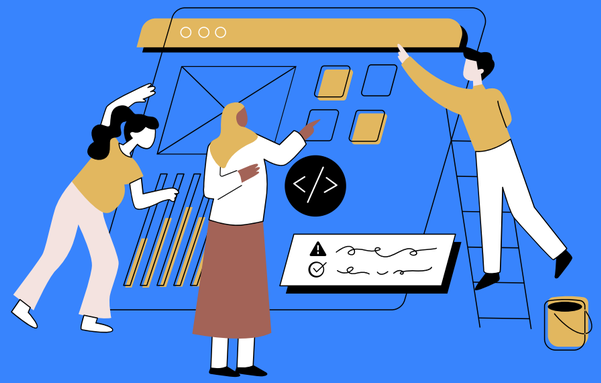
- Introduction
- Operator
- Data Types
- Arrays
- Methods & Classes
- Inheritance
- Package and Interface
- Features of Object-Oriented programming(OOPs)
- Exception Handling
- Multithread programming

Java Applet
String Handling
AWT and AWT Controls
Layout Manager
Happy Ending!



Engineering in One Video (EIOV)

Watch video on YouTube EIOV



History of Web and Internet

History of Web

- Sir Tim Berners-Lee invented the World Wide Web in 1989.
- In the history of the World Wide Web, there are two lines to be traced :
  - The development of hypertext
  - The development of the internet protocols.
- In 1972, DARPA starts research leading to the internet.
- In 1975, Alan Kay produces the first personal computer.
- Berners-Lee developed the first web browser and web server software in 1990, and by 1991, the first web page was created.
- The web quickly grew in popularity as more people began to use it for communication, collaboration, and sharing information.

History of Web

- The introduction of HTML, the markup language used to create web pages, in 1991, and the development of the first web standards in the mid-1990s helped to make the web more accessible and user-friendly.

- The early 2000s saw the rise of web 2.0, a term coined to describe a new generation of web applications that allowed users to contribute and interact with content.

- Social networking sites such as Facebook and Twitter, online video sharing sites such as YouTube, and collaborative platforms such as Wikipedia are all examples of web 2.0 applications.

Engineering in One Video (EIOV)

Watch video on YouTube EIOV

Protocol Governing Web

Protocol Governing Web

- **Hypertext Transfer Protocol (HTTP):**
  - HTTP is the foundation of data communication on the web. It is a protocol used to transfer data over the internet, and it governs how web browsers and servers communicate with each other.

- **Hypertext Transfer Protocol Secure (HTTPS):**
  - HTTPS is a more secure version of HTTP that encrypts data transmitted between a web server and a web browser. It uses SSL (Secure Sockets Layer) or TLS (Transport Layer Security) encryption to protect data from interception and hacking.

- **File Transfer Protocol (FTP):**
  - FTP is a protocol used for transferring files between computers over the internet. It is commonly used to upload and download files from web servers.

Protocol Governing Web

- **Simple Mail Transfer Protocol (SMTP):**
  - SMTP is a protocol used for sending email over the internet. It enables email clients to send messages to mail servers, which then forward the message to the recipient's mail server.

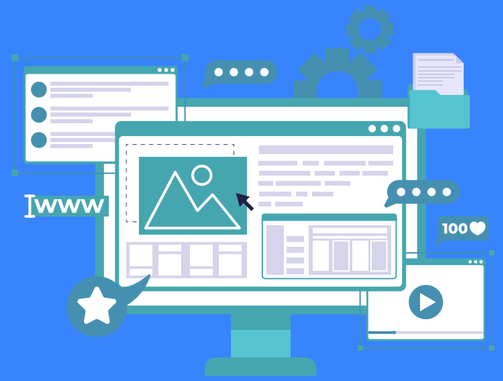
- **Post Office Protocol (POP):**
  - POP is a protocol used for retrieving email from a mail server. It allows email clients to download messages from the mail server and store them locally on the user's computer.

- **Internet Message Access Protocol (IMAP):**
  - IMAP is a protocol used for retrieving and managing email messages. It allows email clients to access messages stored on a mail server and synchronize changes between the server and the client.

- **TCP/IP and UDP**

Engineering in One Video (EIOV)

Watch video on YouTube EIOV



How to develop Web Project

How to develop Web Project

Phase-I : Strategy :

1. Goals and objectives
2. Team building
3. Research and review
4. Project proposal

Phase-II : Design and specification :

1. Developing concepts
2. Content planning
3. Rough design
4. Final design
5. Build prototype
6. Prototype testing

Phase-III : Production or development :

1. Coding

How to develop Web Project

Phase-IV : Testing and maintenance :

1. Automation testing
2. Manual testing

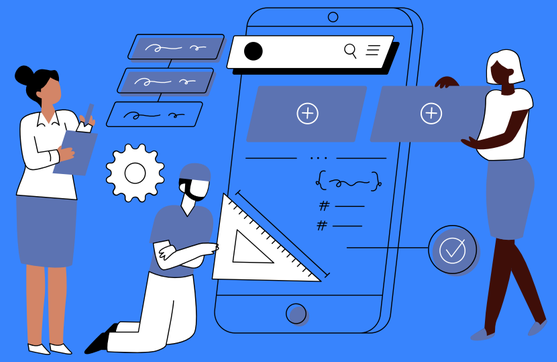
Phase-V : Register with ISP :

1. Buy domain name
2. Hosting

Phase-VI : Launch

Engineering in One Video (EIOV)

Watch video on YouTube EIOV



Internet Services

Internet Services

Internet service provides a way for data to be transferred from internet servers to our computer.

Categories:

- **Communication services**
  - Electronic mail
  - Telnet c. Mailing lists
  - Internet telephony (VoIP)

- **Information retrieval services**
  - FTP

- **Web services:**
  - Allow exchange of information between applications on the web.

- **WWW:**
  - It offers a way to access documents spread over the several servers over the internet.
  - The hyperlinks allow the users to navigate between the documents.

Java

Core Java

Java

Introduction

Core Java

Java is a popular programming language that is widely used for developing desktop, web, and mobile applications.

Some of the key features of Java are:

- **Object-oriented programming:**
  - Allows developers to create modular and reusable code.

- **Platform-independent:**
  - Java code can run on any platform(Windows, Mac, Linux, and other) that has a Java Virtual Machine (JVM).

- **Automatic memory management:**
  - Java uses automatic memory management, which means that developers don't need to manage memory manually. The Java Virtual Machine automatically allocates and deallocates memory.

- **Exception handling:**
  - Allows developers to handle errors and exceptions effectively.

Core Java

- **Rich API**
  - The Java Standard Library includes APIs for file I/O, networking, database access, and more.

- **Multi-threading:**
  - Create applications that can perform multiple tasks simultaneously. Multi-threading can improve application performance and responsiveness.

- **Security:**
  - Java has built-in security features that help protect against viruses, malware, and other security threats.

Engineering in One Video (EIOV)

Watch video on YouTube EIOV

Operator

Engineering in One Video (EIOV)

Watch video on YouTube EIOV

Operator

1. Unary operator :

- The Java unary operators require only one operand.
- Unary operators are used to perform various operations such as :
  - Incrementing/decrementing a value by one (++,--)
  - Inverting the value of a Boolean (!)

2. Arithmetic operator :

- Java arithmetic operators are used to perform addition, subtraction, multiplication, and division. They act as basic mathematical operations.

3. Relational operator :

- Relational operators enables us to test for any relationship between two operands.
- ==(equal to), != (not equal to), > (greater than), < (less than), >= (greater than or equal to), <= (less than or equal to) are the example of relational operator.

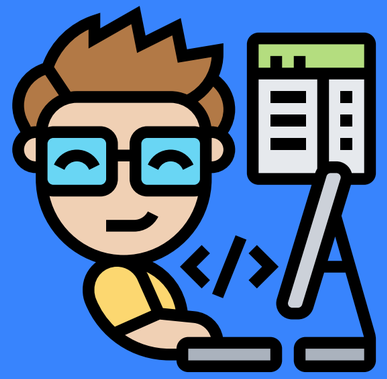
Operator

4. Assignment operator :

- Java assignment operator is one of the most common operators.
- It is used to assign the value on its right to the operand on its left.
- =,+=,-=,*=, /=,%=,|= are the example of assignment operator.

5. Bitwise operator :

- Bitwise operator works on bits and performs bitby-bit operation.
- It can be applied to the integer types, long, int, short, char, and byte.
- Binary AND operator, binary OR operator, binary XOR operator are the example of bitwise operator.



Data Types

Data Types

Primitive Data Types:

- Primitive data types are built-in data types.
- **byte:** an 8-bit signed integer
- **short:** a 16-bit signed integer
- **int:** a 32-bit signed integer
- **long:** a 64-bit signed integer
- **char:** a single 16-bit Unicode character
- **boolean:** a true/false value

Non-Primitive Data Types:

- **String:** a sequence of characters
- **Array:** a collection of elements of the same data type
- **Class:** a blueprint for creating objects
- **Interface:** a collection of abstract methods that can be implemented by classes

These data types can be used to define variables, parameters, and return types in Java methods.

Engineering in One Video (EIOV)

Watch video on YouTube EIOV

Arrays

Arrays

- An array in Java is a data structure that allows you to store a collection of values of the same data type in a contiguous block of memory.
- Each value in the array is identified by an index, which is a non-negative integer that represents its position in the array.
- The first element in the array has an index of 0, the second element has an index of 1, and so on.

```
dataType[] arrayName = new dataType[arraySize];
```

```
int[] myArray = new int[5];
```

```
int[] myArray = {1, 2, 3, 4, 5};
```

Engineering in One Video (EIOV)

Watch video on YouTube EIOV

Methods & Classes

Methods

- In Java, a method is a block of code that performs a specific task and can be invoked (called) by other parts of a program.
- Methods can be used to encapsulate functionality and make code more reusable and easier to maintain.

```
accessModifier returnType methodName(parameterType parameterName) {
    // method body
    return returnValue;
}
```

```
public int add(int a, int b) {
    int sum = a + b;
    return sum;
}
```

```
int result = add(5, 3);
```

Methods

accessModifier: specifies the visibility of the method (e.g., public, private, protected, or default)

returnType: specifies the data type of the value returned by the method (e.g., int, String, boolean, etc.)

methodName: specifies the name of the method

parameterType: specifies the data type of the parameters passed to the method (e.g., int, String, boolean, etc.)

parameterName: specifies the name of the parameters passed to the method

method body: contains the statements that are executed when the method is called
returnValue: specifies the value returned by the method (if any)

Class

- In Java, a class is a blueprint for creating objects that encapsulates data and behavior.
- A class defines the properties (data members) and methods (functions) that all objects of that class will have.

```
accessModifier class ClassName {
    // data members
    dataType memberName;

    // constructors
    ClassName() {
        // constructor body
    }

    // methods
    accessModifier returnType methodName(parameterType parameterName) {
        // method body
        return returnValue;
    }
}
```

Class

- **accessModifier:** specifies the visibility of the class (e.g., public, private, protected, or default)
- **class:** keyword used to define a class
- **ClassName:** specifies the name of the class (by convention, the name should start with a capital letter)
- **data members:** variables that store data within the class
- **constructors:** special methods that are used to initialize objects of the class
- **methods:** functions that define the behavior of the class

Engineering in One Video (EIOV)

Watch video on YouTube EIOV

Inheritance

Engineering in One Video (EIOV)

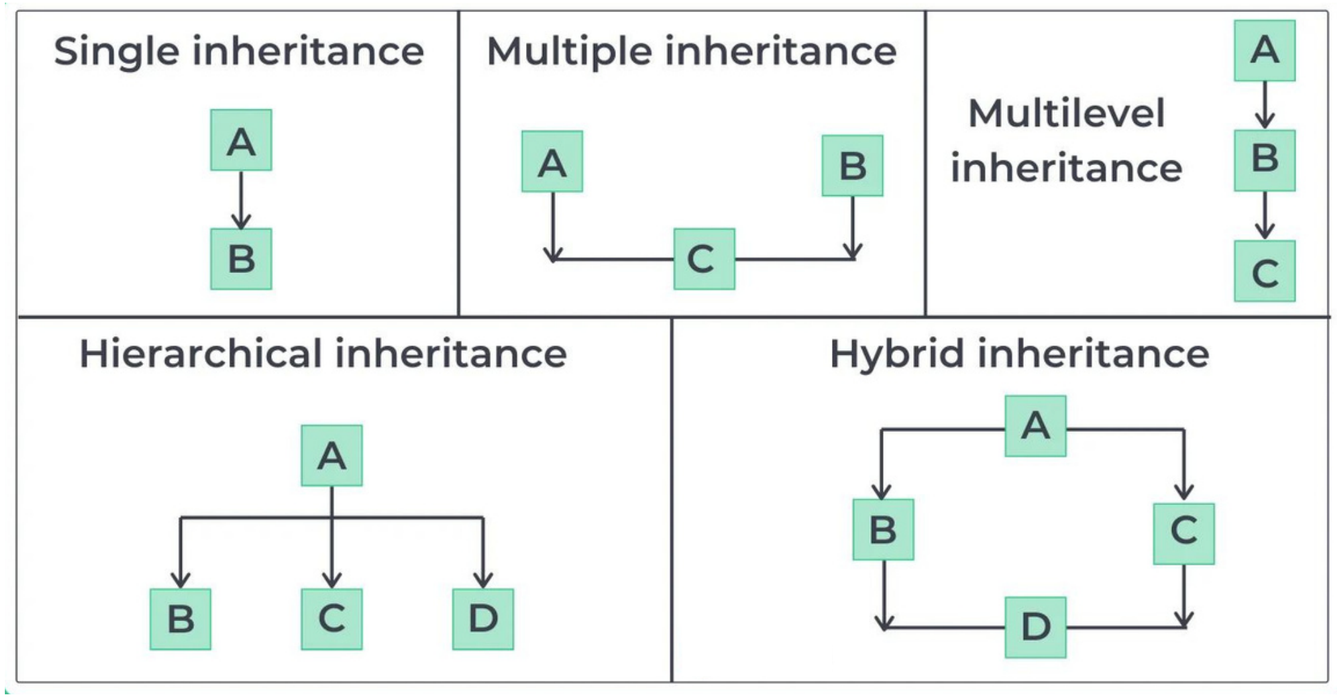
Watch video on YouTube EIOV

Inheritance

- In Java, inheritance is a mechanism that allows one class to inherit the properties and methods of another class.
- The class that is being inherited from is called the superclass or parent class, and the class that is inheriting from it is called the subclass or child class.

- **Types of inheritance :**
  a. **Single inheritance :** It is the inheritance hierarchy wherein one derived class inherits from one base class.
  b. **Multiple inheritance :** It is the inheritance hierarchy wherein one derived class inherits from multiple base classes.
  c. **Hierarchical inheritance :** It is the inheritance hierarchy wherein multiple subclasses inherit from one base class.
  d. **Multilevel inheritance :** It is the inheritance hierarchy wherein subclass acts as a base class for other classes.
  e. **Hybrid inheritance :** The inheritance hierarchy that reflects any legal combination of other four types of inheritance.

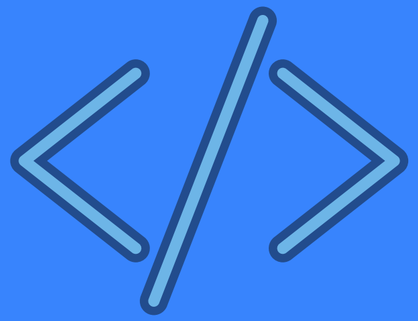
Inheritance



Inheritance

```
accessModifier class SubclassName extends SuperclassName {
    // data members and methods of the subclass
}
```

```
public class Animal {
    public void speak() {
        System.out.println("Animal speaks");
    }
}

public class Dog extends Animal {
    @Override
    public void speak() {
        System.out.println("Woof!");
    }
}
```



Package and Interface

Package

- Package is a mechanism to encapsulate a group of classes, interfaces and subpackages.
- Packages are the way to organize files into different directories according to their functionality, usability as well as category.
- Packages also provide a way for separating “design” from “coding”.

- **There are two types of packages in Java :**
  - **User-defined package :** The package we create is called user defined package.
  - **Built-in package :** The already defined package like java.io.*, java.lang.* etc are known as built-in packages.

Package

```java
package com.example;

public class MyClass {
    // class members
}
```

```java
import com.example.MyClass;

public class AnotherClass {
    MyClass myObject = new MyClass();
}
```

Interface

- An interface defines a set of methods but does not implement them.
- A class that implements the interface agrees to implement all of the methods defined in the interface.
- An interface is a collection of method declarations (without definitions).

```
public interface Drawable {
    void draw();
}
```

- In this example, Drawable is an interface that defines a single abstract method called draw().
- Any class that implements the Drawable interface must provide an implementation of this method.
- To implement an interface in Java, you use the implements keyword followed by the name of the interface.

Features of Object-Oriented programming

Features of Object-Oriented programming

- Objects
- Class
- Inheritance
- Polymorphism
- Data hiding
- Encapsulation
- Reusability

Engineering in One Video (EIOV)

Watch video on YouTube EIOV

Exception Handling

Engineering in One Video (EIOV)

Watch video on YouTube EIOV

Exception Handling

- An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e., at runtime, that disrupts the normal flow of the program's instruction.
- Exception handling provides a type-safe, integrated approach for handling unusual problems that arise while executing a program.
- To handle the exceptions, exception handling mechanism is designed.
- Java provides a mechanism for handling exceptions through the use of try-catch blocks.

Exception Handling

- **Try :**
  - The try block can have one or more statements that could generate an exception.

```
try
{
        statement ;  // generates an exception
}
catch (Exception_type e)
        statement ;  // processes the exception
}
```

  - If any one statement generates an exception, the remaining statements in the block are skipped and execution jumps to the catch block that is placed next to the try block.
  - Every try statement should be followed by at least one catch statement; otherwise compilation error will occur.

Exception Handling

- **Finally:**
  - It can be used to handle an exception that is not caught by any of the previous catch statements.
  - Finally block can be used to handle any exception generated within a try block. c. It may be added immediately after the try block or after the last catch block as follows :

Exception Handling Example

```java
try {
    // The code that might throw an exception goes here
    int result = divide(10, 0); // This will throw an ArithmeticException
    System.out.println(result); // This will not be executed
} catch (ArithmeticException e) {
    // This block will be executed if an ArithmeticException is thrown
    System.out.println("An ArithmeticException occurred: " + e.getMessage());
} finally {
    // This block will always be executed, whether an exception was thrown or not
    System.out.println("This code will always be executed.");
}

// A method that throws an exception
public static int divide(int numerator, int denominator) {
    return numerator / denominator;
}
```

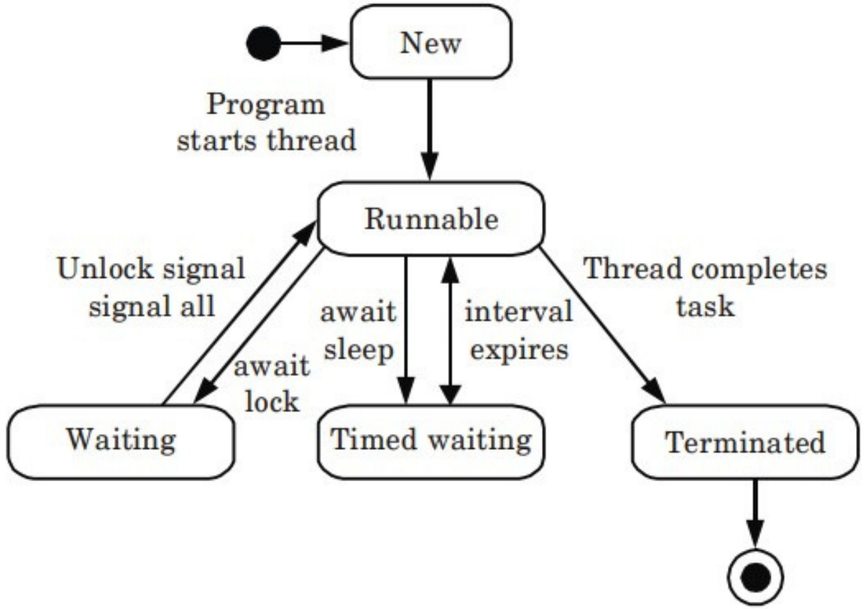
Engineering in One Video (EIOV)

Watch video on YouTube EIOV

Multithread programming

Multithread programming

- A multithreaded program contains two or more parts that can run concurrently.
- Each part of such a program is called a thread, and each thread defines a separate path of execution.
- Each thread runs parallel to each other.
- A multithreading is a specialized form of multitasking.
- In multithread program, each thread has its own life cycle.



Engineering in One Video (EIOV)

Watch video on YouTube EIOV

Java Applet

Java Applet

- An applet in Java is a small program that is designed to be executed within a web browser using the Java Virtual Machine (JVM).
- It runs inside the web browser and works at client side.
- Applets are used to make the website more dynamic and entertaining.

- **Life cycle of an applet use five methods which are as follows :**
  a. **init( )** : This method is intended for whatever initialization is needed for our applet.
  b. **start( )** : This method is automatically called after the browser calls the init method.
  c. **stop( )** : This method is automatically called when the user moves off the page on which the applet sits.
  d. **destroy( )** : This method is only called when the browser shuts down normally.
  e. **paint( )** : Invoked immediately after the start( ) method, and also any time the applet needs to repaint itself in the browser.

Java Applet

```java
import java.awt.Graphics;
import java.applet.Applet;

public class MyFirstApplet extends Applet {
   public void paint(Graphics g) {
      g.drawString("Hello, World!", 50, 25);
   }
}
```

```html
<html>
   <head>
      <title>My First Applet</title>
   </head>
   <body>
      <applet code="MyFirstApplet.class" width="200" height="100"></applet>
   </body>
</html>
```

Engineering in One Video (EIOV)

Watch video on YouTube EIOV

String Handling

String Handling

- String handling is a process of performing different operation such as concatenation, comparison on the string.

- **Java string length() :** The Java string length() method returns the length of the string. It returns count of total number of characters present in the string.
- **compareTo() :** The Java string compareTo() method compares the given string with current string. It returns positive number, negative number or zero.
- **concat() :** The Java string concat() method combines a specific string at the end of another string and returns a combined string.
- **replace() :** replace() method returns a string, replacing all the old characters to new characters. e. Java string equals() : The Java string
- **equals() :** equals() method compares the two given strings on the basis of content of the string. If all the characters are matched, it returns true else it will return false.
- **contains() :** The Java string contains() method searches the sequence of characters in the string. If the sequences of characters are found, then it returns true otherwise returns false.

Engineering in One Video (EIOV)

Watch video on YouTube EIOV

AWT and AWT Controls

AWT and AWT Controls

- AWT stands for Abstract Window Toolkit, which is a Java-based GUI (Graphical User Interface) library that provides the tools and components needed to create desktop applications with a graphical user interface.
- AWT provides a set of classes and interfaces that allow developers to create and manage windows, buttons, text fields, menus, and other graphical user interface components.
- **It contains three kinds of classes :**
  a. **Containers class :** Frame, Dialog, Panel, Applet etc.
  b. **Components class :** TextField, Button, Checkbox, Scrollbar, Label, List etc.
  c. **Custom graphics class :** Color, Font, Dimensions etc.

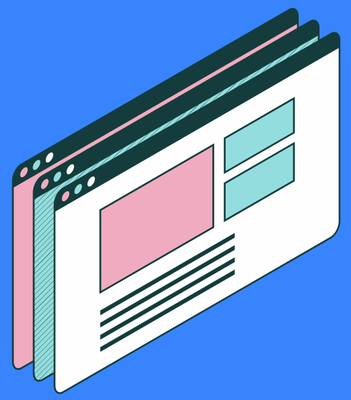
- AWT Controls refer to the set of GUI components that can be created using AWT.
- These components include buttons, checkboxes, choice lists, labels, text fields, text areas, and more.
- AWT Controls are implemented as Java classes and can be instantiated and customized to fit the specific needs of a given application.

AWT and AWT Controls

```java
import java.awt.*;

public class SimpleAWTExample {

    public static void main(String[] args) {
        Frame myFrame = new Frame("My AWT Window");
        Label myLabel = new Label("Hello, World!");
        myFrame.add(myLabel);
        myFrame.setSize(200, 100);
        myFrame.setVisible(true);
    }

}
```

Engineering in One Video (EIOV)

Watch video on YouTube EIOV



Layout Manager

Layout Manager

- Layout manager is a mechanism that automatically arranges the components in a container (such as a Frame, Panel, or Window) according to a specific layout strategy.
- Layout managers are used to control the size and position of components within a container, and to ensure that the components are displayed in a visually pleasing and organized way.
- **There are several layout managers available in Java, including:**
  - **FlowLayout -** arranges components in a single row or column, and wraps them to a new row or column when there is no more space.
  - **BorderLayout -** arranges components in five regions: north, south, east, west, and center.
  - **GridLayout -** arranges components in a grid of rows and columns.
  - **CardLayout -** allows multiple components to be displayed in the same space, and allows the user to switch between them.
  - **GridBagLayout -** provides the most flexible and powerful layout mechanism, allowing components to be placed in a grid of rows and columns with variable sizes and alignments.

Layout Manager

```java
import java.awt.*;
import javax.swing.*;

public class SimpleFlowLayoutExample {

    public static void main(String[] args) {
        JFrame frame = new JFrame("FlowLayout Example");
        JPanel panel = new JPanel(new FlowLayout());

        JButton button1 = new JButton("Button 1");
        JButton button2 = new JButton("Button 2");
        JButton button3 = new JButton("Button 3");

        panel.add(button1);
        panel.add(button2);
        panel.add(button3);

        frame.add(panel);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);
        frame.setVisible(true);
    }

}
```

Engineering in One Video (EIOV)

Happy Ending!

Watch video on YouTube EIOV