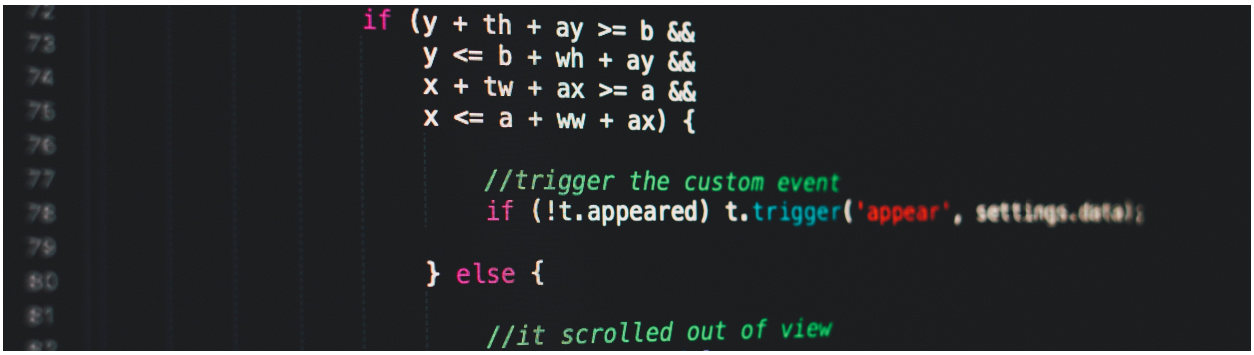


Bit Rebels

OOP’s

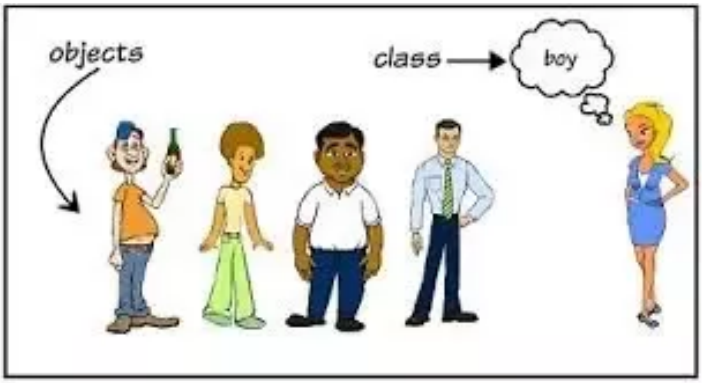
Ye wale oops se confuse na ho…

OOP stands for **Object-Oriented Programming.**

Object-oriented programming has several advantages :

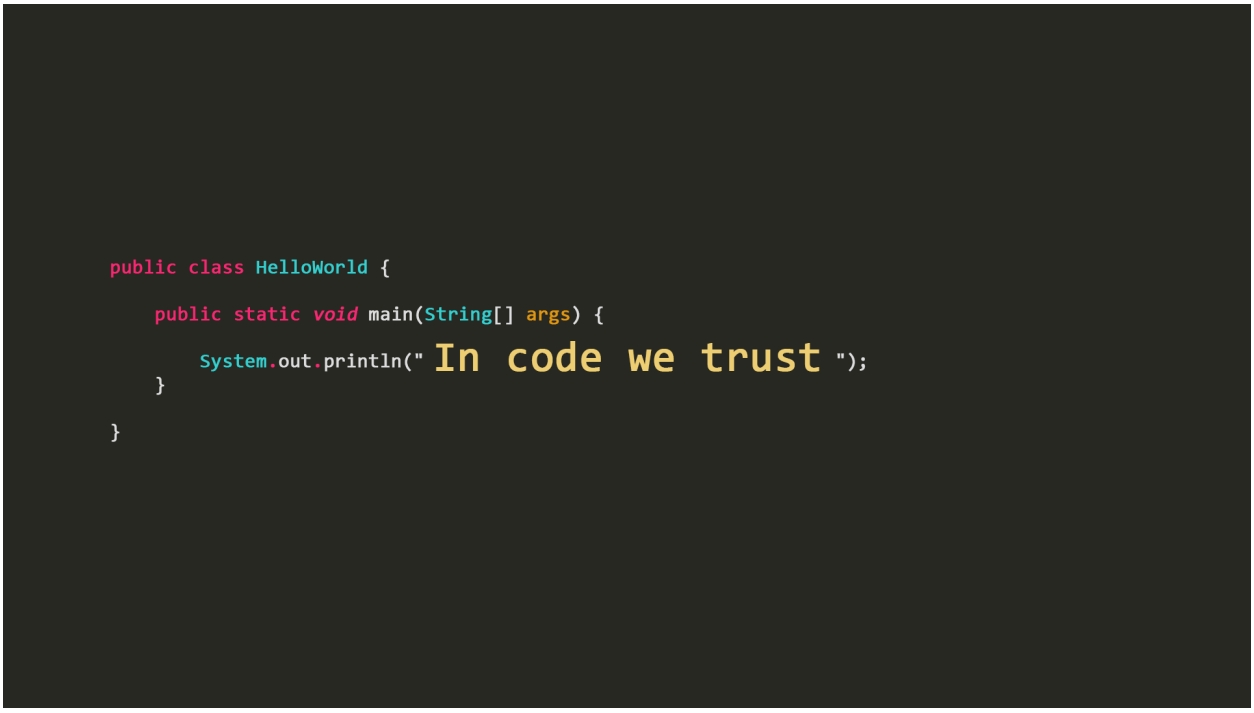
- OOP is faster and easier to execute
- OOP provides a clear structure for the programs
- OOP helps to keep the Java code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug
- OOP makes it possible to create full reusable applications with less code and shorter development time

What are Classes and Objects?



- A class is a template for objects, and an object is an instance of a class; blueprint for instance.
- When the individual objects are created, they inherit all the variables and methods of the class.
- Everything in Java is associated with classes and objects, along with its attributes and methods.

CREATE A CLASS



To create a class, use the keyword `class`:

```
public class Main {
  int x = 5;
}
```

CREATE AN OBJECT

- In Java, an object is created from a class. We have already created the class named `Main`, so now we can use this to create objects.

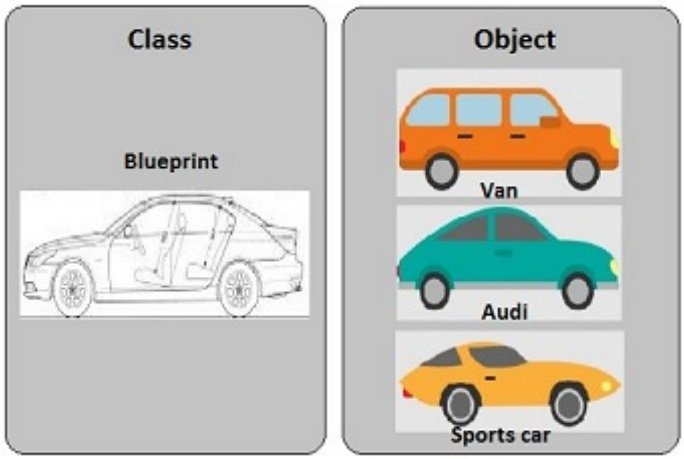
```
public static void main(String[] args) {
    Main myObj = new Main();
    System.out.println(myObj.x);
  }
}
```

MULTIPLE OBJECTS

```
public class Main {
  int x = 5;

  public static void main(String[] args) {
    Main myObj1 = new Main();  // Object 1
    Main myObj2 = new Main();  // Object 2
    System.out.println(myObj1.x);
    System.out.println(myObj2.x);
  }
}
```

USING MULTIPLE CLASSES




```
public class Main {
  int x = 5;
}

class Second {
  public static void main(String[] args) {
    Main myObj = new Main();
    System.out.println(myObj.x);
  }
}
```

MODIFY ATTRIBUTES

```
public class Main {
  int x;

  public static void main(String[] args) {
    Main myObj = new Main();
    myObj.x = 40;
    System.out.println(myObj.x);
  }
}
```

```
public class Main {
  int x = 10;
//final int x = 10;

  public static void main(String[] args) {
    Main myObj = new Main();
    myObj.x = 25;
// x is now 25
    System.out.println(myObj.x);
  }
}
```

PUBLIC V/S STATIC

- A `static` method means that it can be accessed without creating an object of the class.
- A `public` method can only be accessed by objects.

```
public class Main {
  // Static method
  static void myStaticMethod() {
    System.out.println("Static methods can be called without creating objects");
  }
```

```
  // Public method
  public void myPublicMethod() {
    System.out.println("Public methods must be called by creating objects");
  }

  // Main method
  public static void main(String[] args) {
    myStaticMethod(); // Call the static method
    // myPublicMethod(); This would compile an error

    Main myObj = new Main(); // Create an object of Main
    myObj.myPublicMethod(); // Call the public method on the object
  }
}
```

FUNCTIONS

- A **method** is a block of code which only runs when it is called.
- Methods are used to perform certain actions, and they are also known as **functions.**
- Why use methods? To reuse code: define the code once, and use it many times.

TYPES

- User defined methods
- Pre-defined methods

User Defined Method

- These methods are those which are defined by the user or in our case programmer.
- The programmer can manipulate the working of the functions.

Pre-defined Method

- These methods are defined in libraries and packages.
- The segment work is predefined and is meant for a specific work.
- A user can’t manipulate the definition and the prototype of the method.

CREATE METHOD

- A method must be declared within a class. It is defined with the name of the method, followed by parentheses **()**

```
public class Main {
  static void myMethod() {
    // code to be executed
  }
}
```

CALL A METHOD

```
public class Main {
  static void myMethod() {
    System.out.println("I just got executed!");
  }

  public static void main(String[] args) {
    myMethod();
  }
}
```

RETURN VALUE

```
public class Main {
  static int myMethod(int x) {
    return 5 + x;
  }

  public static void main(String[] args) {
    System.out.println(myMethod(3));
  }
}
```

PARAMETERS AND ARGUMENTS

PARAMETERS:

- Information can be passed to methods as parameters.
- Parameters act as variables inside the method.
- Parameters are specified after the method name, inside the parentheses.

```
public class Main {
  static void myMethod(String fname) {
    System.out.println(fname + " Refsnes");
  }

  public static void main(String[] args) {
    myMethod("Liam");
    myMethod("Jenny");
    myMethod("Anja");
  }
}
```

```
public class Main {
  static void myMethod(String fname, int age) {
    System.out.println(fname + " is " + age);
  }

  public static void main(String[] args) {
    myMethod("Liam", 5);
    myMethod("Jenny", 8);
    myMethod("Anja", 31);
```

```
  }
}
```

SCOPE RESOLUTION

- In Java, variables are only accessible inside the region they are created. This is called **scope.**

```
public class Main {
  public static void main(String[] args) {
    // Code here CANNOT use x
    int x = 100;
    // Code here can use x
    System.out.println(x);
  }
}
```

```
public class Main {
int x = 100;
  public static void main(String[] args) {
    // Code here can use x
    System.out.println(x);
  }
}
```

FLOW OF CONTROL

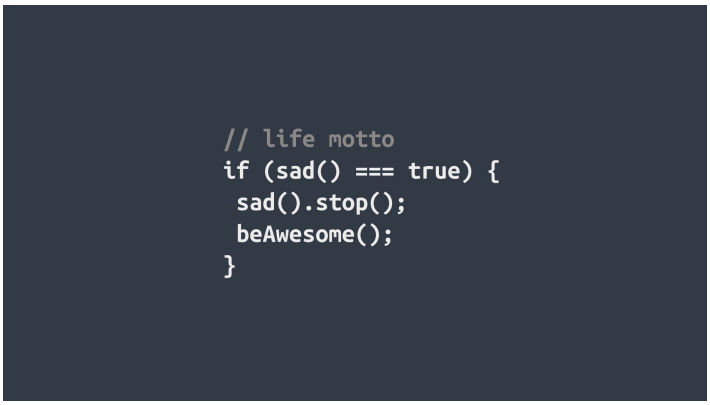
- JAISE PAANI BEHTA HAI WAISE

CONDITIONAL STATEMENT

- In computer science, conditionals are programming language commands for handling decisions.
- Conditionals perform different computations or actions depending on whether a programmer-defined boolean condition evaluates to true or false.

TYPES

IF…ELSE STATEMENT



- Use the `if` statement to specify a block of Java code to be executed if a condition is `true`

```
if (condition) {
  // block of code to be executed if the condition is true
}
```

```
if (20 > 18) {
  System.out.println("20 is greater than 18");
}
```

```
int x = 20;
int y = 18;
if (x > y) {
  System.out.println("x is greater than y");
}
```

- Use the `else` statement to specify a block of code to be executed if the condition is `false`

```
if (condition) {
  // block of code to be executed if the condition is true
} else {
  // block of code to be executed if the condition is false
}
```

```
int time = 20;
if (time < 18) {
  System.out.println("Good day.");
} else {
  System.out.println("Good evening.");
}
```

- Use the `else if` statement to specify a new condition if the first condition is `false`

```
if (condition1) {
  // block of code to be executed if condition1 is true
} else if (condition2) {
  // block of code to be executed if the condition1 is false and condition2 is true
} else {
  // block of code to be executed if the condition1 is false and condition2 is false
}
```

```
int time = 22;
if (time < 10) {
  System.out.println("Good morning.");
```

```
} else if (time < 18) {
  System.out.println("Good day.");
} else {
  System.out.println("Good evening.");
}
```

- Experiment

```
int x = 50;
int y = 10;
__(x __ y) {
  System.out.println("Hello World");
}
```

TERNARY OPERATOR

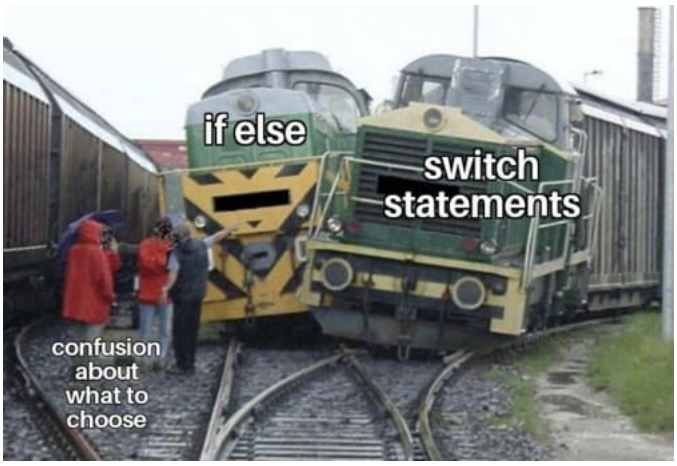
- Short-hand if else
- It consists of three operands

```
variable = (condition) ? expressionTrue :  expressionFalse;
```

```
int time = 20;
if (time < 18) {
  System.out.println("Good day.");
} else {
  System.out.println("Good evening.");
}

int time = 20;
String result = (time < 18) ? "Good day." : "Good evening.";
System.out.println(result);
```

SWITCH CASE STATEMENT



- The `switch` statement selects one of many code blocks to be executed
- It compares the variable with the case value specified. If found true, it will execute the block defined else will either check for other case or will execute the default.

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

This is how it works:

- The `switch` expression is evaluated once.
- The value of the expression is compared with the values of each `case`.
- If there is a match, the associated block of code is executed.
- The `break` and `default` keywords are optional, and will be described later in this chapter

```
int day = 4;
switch (day) {
  case 1:
    System.out.println("Monday");
    break;
  case 2:
    System.out.println("Tuesday");
    break;
  case 3:
    System.out.println("Wednesday");
    break;
  case 4:
    System.out.println("Thursday");
    break;
  case 5:
    System.out.println("Friday");
    break;
  case 6:
    System.out.println("Saturday");
    break;
  case 7:
    System.out.println("Sunday");
    break;
}
```

- **BREAK**: This will stop the execution of more code and case testing inside the block

> 💡 A break can save a lot of execution time because it "ignores" the execution of all the rest of the code in the switch block.

- **FALL THROUGH CONDITION**


```
int day = 4;
switch (day) {
  case 1:
    System.out.println("Monday");
  case 2:
    System.out.println("Tuesday");
    break;
  case 3:
    System.out.println("Wednesday");
    break;
  case 4:
    System.out.println("Thursday");
  case 5:
    System.out.println("Friday");
  case 6:
    System.out.println("Saturday");
    break;
  case 7:
    System.out.println("Sunday");
    break;
}
```

- **DEFAULT**: The `default` keyword specifies some code to run if there is no case match

```
int day = 4;
switch (day) {
  case 6:
    System.out.println("Today is Saturday");
    break;
  case 7:
    System.out.println("Today is Sunday");
    break;
  default:
    System.out.println("Looking forward to the Weekend");
}
```

ITERATIVE STATEMENTS

- Iteration statements **cause statements (or compound statements) to be executed zero or more times, subject to some loop-termination criteria**
- Loops can execute a block of code as long as a specified condition is reached.
- Loops are handy because they save time, reduce errors, and they make code more readable.

TYPES

FOR LOOP

- When you know exactly how many times you want to loop through a block of code, use the `for` loop

```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
}
```

Statement 1 is executed (one time) before the execution of the code block.

Statement 2 defines the condition for executing the code block.

Statement 3 is executed (every time) after the code block has been executed.

```
for (int i = 0; i < 5; i++) {
  System.out.println(i);
}
```

```
for (int i = 0; i <= 10; i = i + 2) {
  System.out.println(i);
}
```

- **NESTED LOOP:**

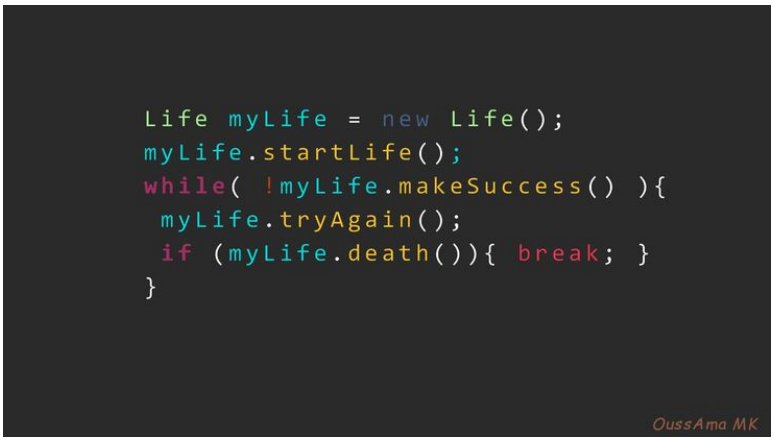
```
// Outer loop
for (int i = 1; i <= 2; i++) {
  System.out.println("Outer: " + i); // Executes 2 times

  // Inner loop
  for (int j = 1; j <= 3; j++) {
    System.out.println(" Inner: " + j); // Executes 6 times (2 * 3)
  }
}
```

- **FOR-EACH LOOP:**

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (String i : cars) {
  System.out.println(i);
}
```

WHILE LOOP



- The `while` loop loops through a block of code as long as a specified condition is `true`

```
while (condition) {
  // code block to be executed
}
```

```
int i = 0;
while (i < 5) {
  System.out.println(i);
  i++;
}
```

DO/WHILE LOOP

- The `do/while` loop is a variant of the `while` loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true

```
do {
  // code block to be executed
```

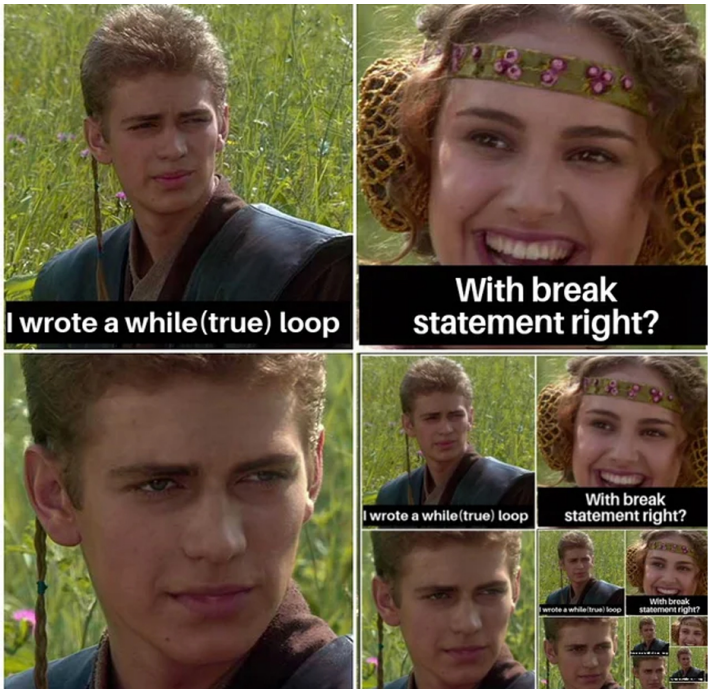
```
}
while (condition);
```

```
int i = 0;
do {
  System.out.println(i);
  i++;
}
while (i < 5);
```

BREAK AND CONTINUE

- The `break` statement can also be used to jump out of a **loop**

```
for (int i = 0; i < 10; i++) {
  if (i == 4) {
    break;
  }
  System.out.println(i);
}
```




```
int i = 0;
while (i < 10) {
  System.out.println(i);
  i++;
  if (i == 4) {
    break;
  }
}
```

- The `continue` statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop

```
for (int i = 0; i < 10; i++) {
  if (i == 4) {
```

```
    continue;
  }
  System.out.println(i);
}
```

```
int i = 0;
while (i < 10) {
  if (i == 4) {
    i++;
    continue;
  }
  System.out.println(i);
  i++;
}
```

RECURSION

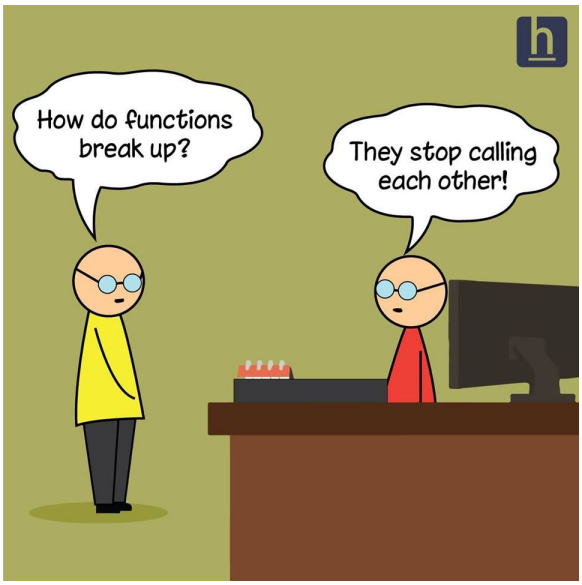
- Recursion is the technique of making a function call itself.
- This technique provides a way to break complicated problems down into simple problems which are easier to solve.

```
public class Main {
  public static void main(String[] args) {
    int result = sum(10);
    System.out.println(result);
  }
  public static int sum(int k) {
    if (k > 0) {
      return k + sum(k - 1);
    } else {
      return 0;
    }
  }
}
```

When the `sum()` function is called, it adds parameter `k` to the sum of all numbers smaller than `k` and returns the result. When k becomes 0, the function just returns 0. When running, the program follows these steps:

```
10 + sum(9)
10 + ( 9 + sum(8) )
10 + ( 9 + ( 8 + sum(7) ) )
...
10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 + sum(0)
10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 + 0
```

HALTING CONDITION/ BASE CONDITION



- Just as loops can run into the problem of infinite looping, recursive functions can run into the problem of infinite recursion
- Infinite recursion is when the function never stops calling itself.
- Every recursive function should have a halting condition, which is the condition where the function stops calling itself.

```
public class Main {
  public static void main(String[] args) {
    int result = sum(5, 10);
    System.out.println(result);
  }
  public static int sum(int start, int end) {
    if (end > start) {
      return end + sum(start, end - 1);
    } else {
      return end;
    }
  }
}
```

💡 The developer should be very careful with recursion as it can be quite easy to slip into writing a function which never terminates, or one that uses excess amounts of memory or processor power. However, when written correctly recursion can be a very efficient and mathematically-elegant approach to programming.

RESOURCE

Java Tutorial

Java is a popular programming language. Java is used to develop mobile apps, web apps, desktop apps, games and much more. Start learning Java now " Our "Try it Yourself" editor makes it easy to learn Java. You can edit Java code and view the result in your browser.

https://www.w3schools.com/java/default.asp

Control GIFs | Tenor

With Tenor, maker of GIF Keyboard, add popular Control animated GIFs to your conversations. Share the best GIFs now >>>

https://www.google.com/url?sa=i&url=https%3A%2F%2Ftenor.com%2Fsearch%2Fcontrol-gifs&psig=AOvVaw1kzvpuz_pLBzdxUsbhufrS&ust=1672860852330000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCNCMy

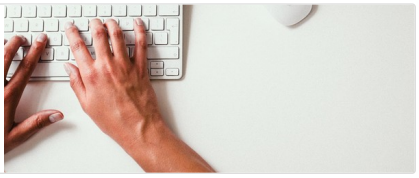
7mSrPwCFQAAAAAdAAAAABAj

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.pinterest.com%2Fpin%2F143059725641565518%2F&psig=AOvVaw0iqKlWHa1QOzrHaE5fEFnT&ust=1672859563119000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCKjV-4mcrPwCFQAAAAAdAAAAABAE

What's the best way to define the words "class" and "object" to someone who hasn't used them?

Object Oriented programming is about creating programs using as building blocks, "things" that exists in the real world, these real world things are called objects, hence object oriented For instance, if you're creating a Address Book program, you may define the following objects: person, address, phone Among many, many others.

https://www.google.com/url?sa=i&url=https%3A%2F%2Fstackoverflow.com%2Fquestions%2F3686647%2Fwhats-the-best-way-to-define-the-words-class-and-object-to-someone-who-hasn&psig=AOvVaw29pZo-zD4k-rrpbNTcc09U&ust=1672858009761000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCIC966WcrPwCFQAAAAAdAAAAABAD



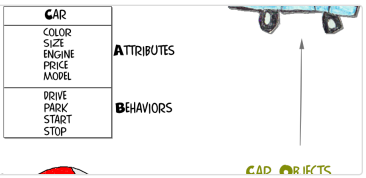
What is the concept of object and classes?

Answer (1 of 13): Class is the generic category to which you associate certain objects. For example, you associate a Pomerian or Labrador or Golden Retriever to Dog class. They exhibit certain distinctive features from other animals (say cats) which helps us to categorize them into one single cla...

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.quora.com%2FWhat-is-the-concept-of-object-and-classes&psig=AOvVaw29pZo-zD4k-rrpbNTcc09U&ust=1672858009761000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCIC966WcrPwCFQAAAAAdAAAAABAT

Object Oriented Programming: Classes and Objects

We do things differently! BHiveTech blog is not just another site to find information about technology and the software world. It is also a place to have fun while learning new cool stuff about the tech world. Get on your seat and hold your breath... this journey is about to start!



https://www.google.com/url?sa=i&url=https%3A%2F%2Fbhivetech.blog%2Fobject-oriented-programming-classes-and-objects%2F&psig=AOvVaw29pZo-zD4k-rrpbNTcc09U&ust=1672858009761000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCIC966WcrPwCFQAAAAAdAAAAABAh

Quotes about Java programming language (19 quotes)

However, when Java is promoted as the sole programming language, its flaws and limitations become serious.

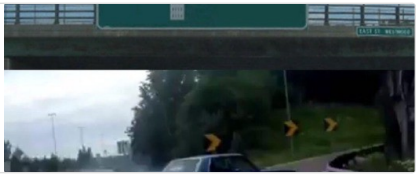
https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.quotemaster.org%2Fjava%2Bprogramming%2Blanguage&psig=AOvVaw0HPLhiuu9lMVlfcxfbY2Em&ust=1672863784464000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCIDzsq-drPwCFQAAAAAdAAAAABAK

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.pinterest.com%2Fpin%2F705446729133735066%2F&psig=AOvVaw0HPLhiuu9lMVlfcxfbY2Em&ust=1672863784464000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCIDzsq-drPwCFQAAAAAdAAAAABAQ

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.pinterest.com%2Fpin%2F775674735804387672%2F&psig=AOvVaw0HPLhiuu9lMVlfcxfbY2Em&ust=1672863784464000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCIDzsq-drPwCFQAAAAAdAAAAABAe

What's a Ternary Operator?

In programming, we use conditional logic for almost every problem we try to solve. Meaning that if something happens then do this, if not either proceed as is or do something else. This concept drives a lot of the logic in our applications but can sometimes a headache to track and manage.



https://www.google.com/url?sa=i&url=https%3A%2F%2Fmedium.com%2Fmatts-lambda-minutes%2Fwhats-a-ternary-operator-b87d576f92f1&psig=AOvVaw0ufXHrNRel_3qf9t1gyrCe&ust=1672864221808000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCMCdjoCfrPwCFQAAAAAdAAAAABAQ

If-Else or Switch-Case: Which One to Pick?

This article was originally published on Dasha In case you are wondering, Dasha is a conversational-AI-as-a-service platform that lets you embed realistic voice and text conversational capabilities into your apps or products. Start building for free! If you are a newbie to programming and JavaScript, you might be confused

https://www.google.com/url?sa=i&url=https%3A%2F%2Fdev.to%2Fsumusiriwardana%2Fif-else-or-switch-case-which-one-to-pick-4p3h&psig=AOvVaw1tFmG6aYI7WL3T0pbo-OhM&ust=1672864301757000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCKCcj6afrPwCFQAAAAAdAAAAABAe

Code Memes on Twitter: "Do while loops...!https://t.co/jvQrLZfoVV#loops #dowhile #while #programming #forloop #programmer #programmerslife #progra

Do while loops...!https://t.co/jvQrLZfoVV#loops #dowhile #while #programming #forloop #programmer #programmerslife #programmershumor #programmersmemes #program

https://www.google.com/url?sa=i&url=https%3A%2F%2Ftwitter.com%2Fcode_memez%2Fstatus%2F1336604129154670592&psig=AOvVaw1gaa1UIiWRvsFr2Aq3rjw3&u

Log in or sign up to view

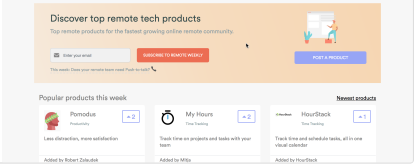
See posts, photos and more on Facebook.

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.facebook.com%2Floopofmemes%2Fphotos%2Fa.107594584527192%2F177179334235383%2F%3Ftype%3D3&psig=AOvVaw1gaa1UIiWRvsFr2Aq3rjw3&ust=1672864720609000&source=images&cd=vfe&ved=0CBAQjRxqGAoTCMDs7u2grPwCFQAAAAAdAAAAABCjAQ

65 Best, Super Funny Programmer Memes to Crack You Up

If you're a programmer, then you've probably had a busy day writing code. Be it for your own product or to meet client demands. And as much as you love it, it can get a bit too much at times. We understand, and have hence put together these programmer memes for you.

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.remote.tools%2Fmemes%2Fprogrammer-memes&psig=AOvVaw0PIOHVw-ws0STu-XMkgrLt&ust=1672865065548000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCOCtv5KirPwCFQAAAAAdAAAAABAW



i need a break in my switch statement

i need a break in my switch statement Programmer

https://www.google.com/url?sa=i&url=http%3A%2F%2Fm.quickmeme.com%2Fmeme%2F35l13s&psig=AOvVaw0PIOHVw-ws0STu-XMkgrLt&ust=1672865065548000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCOCtv5KirPwCFQAAAAAdAAAAABAd

https://www.google.com/url?sa=i&url=https%3A%2F%2Fme.me%2Ft%2Ffunction&psig=AOvVaw182zIYSxAUSvmUQ44k_neP&ust=1672865288474000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCJCW0fyirPwCFQAAAAAdAAAAABAJ

THANK YOU

- Hope so we tried to delevier as much content that we can and you grasped that easily…