OBJECT ORIENTED PROGRAMMING USING

Java

Let's explore technology together to live in the future

Checkout more on https://github.com/Sy-hash-collab

Sy-hash-collab

OOP - Object Oriented Programming with JAVA

Object oriented programming is about creating object that contain both data and methods.
It is a programming technique in which the programs are written on the basis of objects
Classes and Objects are two main aspects.

Features of OOP

Objects:

OOP provides the facility of programing based on objects. Object is an entity that consist of data and functions.
An object is an entity in real world such as person, thing, concept etc., identified by its name
It consist of:

- Properties - which are charachteristis of object
- Functions - which are the action that can be performed by an object.

Examples → Physical Objects - vehicles, car, bus
Examples → Userdefined datatypes - Time, Angles

Properties of Objects:

The charachteristics of an object are known as its properties/attributes
Each object has its own properties, can be used to describe the object.
Properties of a Person and a Car can b

Person	Car
Name	Color
Age	Price
Weight	Model

Functions of Object:

An object can perform different tasks and actions. The actions that can be performed by an object are known as functions / methods.
For example, the object car can perform function
- Start
- Stop
- Accelerate
- Reverse

The set of all function an object represent the behavior of object. The overall behaviour of an object can be determined by the list of functions of that object.

• Classes:

A collection of objects with same properties and functions is known as class.
A class is used to define the charachteristics of objects.
=> For example, a class Person can be used to define the charachteristics and functions of a person. It can be used to create many objects of type Person such as Ali, Usman, Ahmed. All objects of Person class will have same charachteristis and functions.
=> The values of each object can be different.
=> Each object of a class is known as an instance of its class. For example, Ali, Usman, Ahmed are three instances of a class "Person", similarly myBook, yourBook can be two instances of class "Book".

JAVA

Multiplatform Programming Language: Java is MPL and can work on different operating systems.
Opensource Programming Language: Functionality of Java can be enhanced and modified.
IDE → • Netbeans • Eclipse

• Case - Sensitive language.

Java final keyword:

The final keyword is a non-access modifier used for classes, attributes and methods, which makes them non-changeable (impossible to inherit or override).
The final keyword is useful when you want a variable to always store same value like PI

```
final int x = 10;
```

Java Strings:

Strings are used for storing text
A String variable contains a collection of charachters surrounded by double quotes:

```
String greeting = "Hello";
```

* Keywords are always in lowercase inJava
* A word is a class in Java whose first letter is in uppercase

Java - What are Classes & Objects:

- So, class is a tempelate for objects,
- an object is an instance of a class.

Class	Objects
1) Fruit	· Apple · Banana · Mango
2) Car	· Volvo · Audi · Toyota

Student → ClassName

$\left.\begin{matrix}\text{rollno.}\\\text{marks}\end{matrix}\right\}\rightarrow$ Data Members, Attributes, Properties

$\left.\begin{matrix}\text{attendance()}\\\text{result()}\end{matrix}\right\}\rightarrow$ Methods, Memberfunction.

What is Java?

Java is general purpose, class based, object oriented programming language which works on different operating systems such as Windows, Mac, Linux.

In Java, every application starts with a classname, and this class must match the file name. When saving a file, save it using the class name and add .java to end of name.

Let's write a program that prints the message "Hellofree Code Camp". We're going to start our first java file called Main.java

```
public class Main
{
    public static void main (String[] args)
    {
        System.out.printIn ("Hello free Code Camp")
    }
}
```

* Every line of code that runs in Java must be in a class. * Java is case-sensitive.
Let's see what that code's doing:
=> First look at the main() method:
public static void main (String [] args).
• This method is required in every Java program, it's most important because it's the entry point of any Java Program.
• Its syntax is always same. The only thing that can be changed is the name of string array argument. For example, you can change args to myStringArgs.

What is a Class in Java?

A class is defined as "collection of objects". You can also think of a class as a blueprint from which you can create an individual object.
• To create a class, we use the keyword class

```
class ClassName
{   //fields
    //methods
}
```

*a class should always start with uppercase first letter.

• In the above syntax, we have fields (also called variables) and methods, which represent the state and behaviour of object.
Note that in Java we use:
• fields to store data.
• methods to perform operations.

```
public class Main {
    int y= 2;
}
```

What is an Object in Java?

An object is an entity in the real world that can be distinctly identified. Objects have methods and properties to make a particular type of data useful.

- **A unique identity:** Each object has a unique identity, evenif the state is identical to that of another object.
- **Properties / Attributes / State / Data Members:** State tells us how the object looks or what properties it has.
- **Behaviour / Methods / Member function:** Behaviour tells us what the object does.

	Ex.1	Ex.2
Object	Car	House
State	color, brand, model	adress, location
Behaviour	break, accelerate, turn, changegears	open door, close door

```
public class Number {

  int y = 10;

 public static void main (String[] args)
 {
    Number myObj = new Number();

    System.out.printIn (myObj.y);
 }
}
```

Most programming languages allow you to store and keep track of three distinct types of data i-e text, number, boolean values.
We can represent all different types of info. with these datatypes.
But what happens when we need to work with more complex data, we want to represen something more complex lik entity in realwork
Let's take a student for example, imagine we're writing a program for a school and we wanted to keep track of students who're currently enrolled, keep tracking of names with strings or prices with numbers, we want to be able to keep track of students
The problem is we don't have a datatype for students, there's no student datatype that built into the language like string numbers
We need more complex datatypes, so we can use object-oriented programming.
OOP allows developes to create their own datatypes. So that a developer could create the student datatype and then use it like they would a normal string number.
So how can we go about creating our student datatype?
First step is to create a student class, in oop a Class is specification of datatype it's a blueprint that tells the programming language what the new datatype looks like.
In our Student class, we can define attributes and functionality that make up a student. string, numbers.

In this case, we say, a student might have a name attribute which would be a string, a GPA attribute which would be a number a year attribute and maybe it has scholarship boolean which would determine whether or not the student has a scholarship. so, you can define all these different attributes of student which would be either strings, numbers, bolean. We can also define functionality around Student so we might create a hasHonors() function inside of student class which woulduse the GPA of student to determine if they qualify forHonors(), giveScholarship() function which would set the hasScholarship boolean to true, so class can have attributes, functionality.

Student

name (string)
gpa (number)
year (string)
has Scholarship (boolean)

} Attributes

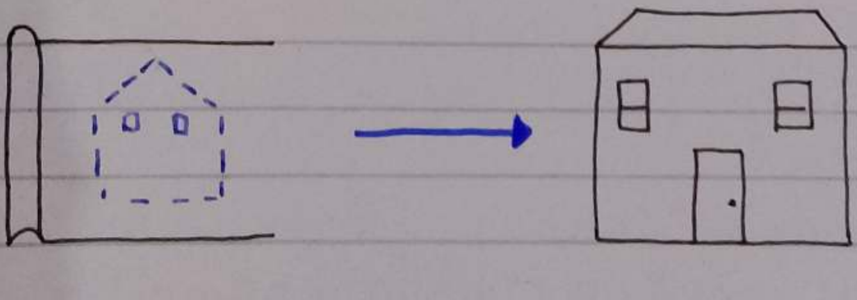
hasHonors () (func.)
giveScholarship() (func.)

} Functions

When we create a class we tell language what this new datatype looks like, what it does but we're doing here is we're not actually creating a student that we can work with, to create a student in our program we'll need an object.

We'll need an object, an object is an instance of a class which means a student would be an instance of our student class. So, a Student object is an actual student with an actual name, GPA, year and scholarship status. So, in our program we might create three student objects with names Jack, Kate, Sawyer and each of them would be an instance of our student class. Now, the object is actual thing that we're gonna work with in our program's, you could pass it around, you can store it inside of a variable you could access each of those attributes or call any of those functions.
Object is an implementation of class.
Class is simply a blueprint.

Student Object 1	Student Object 2
Kate	Jack
3.4	2.6
Sophomore	Senior
True	False
has Honors()	hasHonors()
give Scholarship()	giveScholarship()



Class Object

Java Variables:
Variables are containers for storing data values.

- String - stores text
- int - stores integers.
- float - stores floating point numbers
- char - stores single charachters
- boolean - stores values with twostates: true/ false.

type variablename = value;

Ex:
```
1) String name = "John";
   System.out.println(name);
2) int myNum = 15;
   System.out.println(myNum);
```

• Final Variables:
use the "final" keyword, this will declare the variable as "constant", which means unchangeable.
```
final int myNum = 15;
 myNum = 20; // will generate error.
```

• Display Variables:
The println() method is used to display variables.
To combine both text and a variable use the '+' charachter.
```
String name = "John";
System.out.println("Hello" + name);
```

Identifiers:

All java variables mustbe identifie with unique names. These names are called identifiers.
They can be shortnames (x,y) or descriptive names (age, sum).

```
int minutesPerHour = 60;
int m = 60;
```

Data Types

Primitive	Non-primitive
byte, short, int, long float, double, char boolean	String, Arrays and Classes.

The main differences b/w two datatypes:

• Pre-defined (already defined) in Java	• Created by programmer not defined by Java.
• Cannot be used	• Can be used to call method. to perform certain operations
• Always has a value	• Can be null.
• Starts with lowercase	• Starts with uppercas letter.

Java Output / Print:

You can use the println() method to output values.

```
System.out.println ("Hello, World");
```

Scienticfic Numbers:

A floating point number can also be a scientific number with an 'e' to indicate power of 10.

```
float f1 = 35e3f;
double d1 = 12E4d;
System.out.println (f1);
System.out.println (d1);
```

Java Comments:

Explain Java code, make it more readable, can be used to prevent execution when testing code.

- Single-line coments:

Singleline comments start with two forward slashes (//).

- Multi-line coments:

Multiline coments start with /* and ends with */

Boolean Types:

You'll need a datatype that can only have one of two values:

YES / NO   TRUE / FALSE   ON / OFF

```
boolean isJavaFun = true;
boolean isFishTasty = false;

System.out.println (isJavaFun); output true
System.out.println (isFishTasty); output false
```