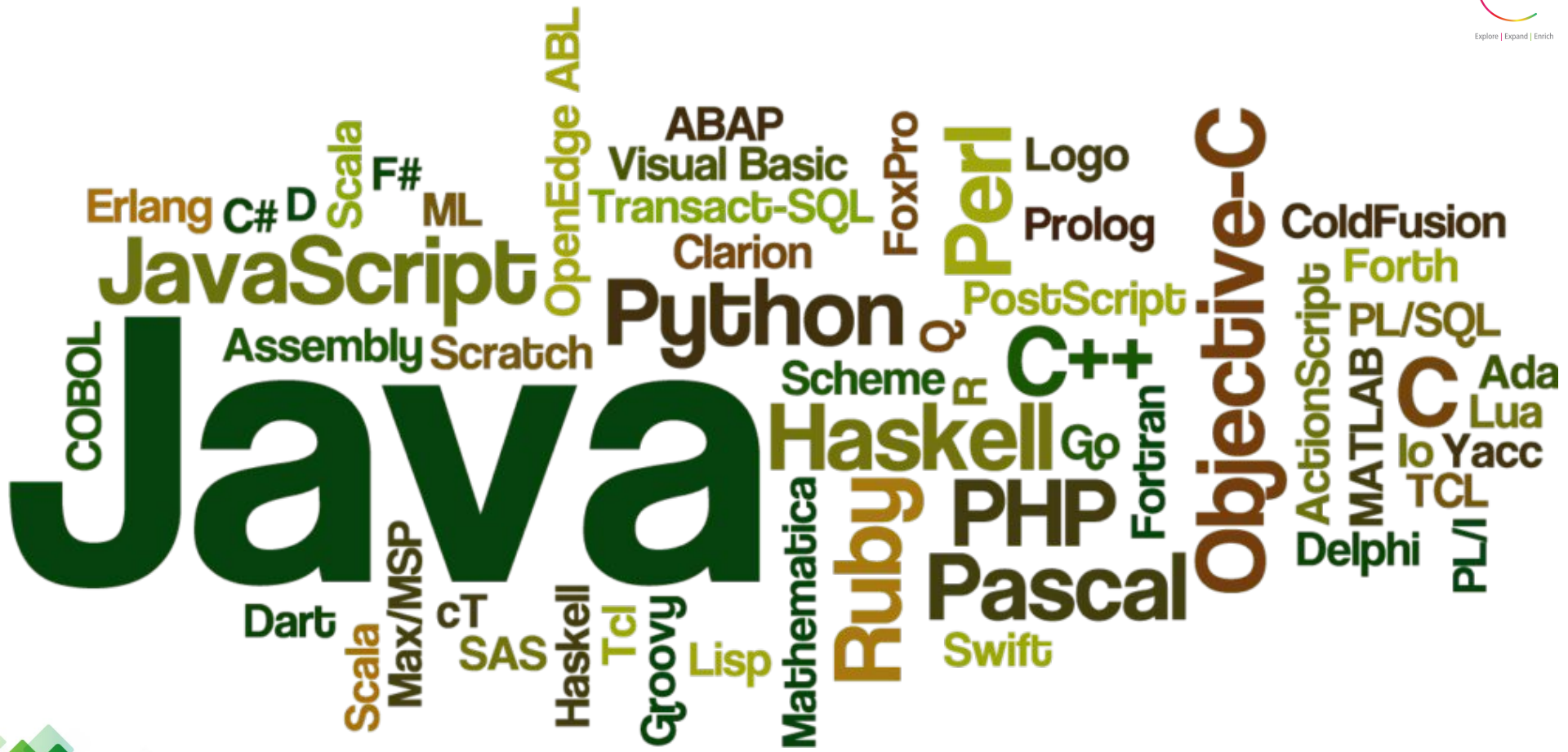
Explore | Expand | Enrich



INHERITANCE IN JAVA

INTRODUCTION

- The process by which one class acquires the properties(data members) and functionalities(methods) of another class is called **inheritance**
- When a Class extends another class it inherits all non-private members including fields and methods
- Inheritance in Java can be best understood in terms of Parent and Child relationship, also known as **Super class**(Parent) and **Sub class**(child)
- Inheritance defines **is-a relationship** between a Super class and its Sub class
- The **extends** keywords are used to describe inheritance in Java

INHERITANCE

Child Class:

The class that extends the features of another class is known as child class, sub class or derived class

Parent Class:

The class whose properties and functionalities are used(inherited) by another class is known as parent class, super class or Base class

SYNTAX

```
class Subclass-name extends Superclass-name
{
   //methods and fields
}
```

- The **extends keyword** indicates that you are making a new class that derives from an existing class
- The meaning of "extends" is to increase the functionality

EXAMPLE

```
class Vehicle {
 ......
}
 class Car extends Vehicle {//extends the property of vehicle
class
.......
 }
```

PURPOSE

- It promotes the code reusability i.e the same methods and variables which are defined in a parent/super/base class can be used in the child/sub/derived class
- It promotes polymorphism by allowing method overriding

DISADVANTAGES

- Main disadvantage of using inheritance is that the two classes (parent and child class) gets **tightly coupled**

- This means that if we change code of parent class, it will affect to all the child classes which is inheriting/deriving the parent class, and hence, **it cannot be independent of each other**

EXAMPLE

```
class Parent {
     public void p1() {
     System.out.println("Parent method");
     }
}
public class Child extends Parent {
     public void c1() {
     System.out.println("Child method");
     }
     public static void main(String[] args)
{
     Child cobj = new Child();
     cobj.c1();
     cobj.p1();
     }
}
```

EXAMPLE

```
class Vehicle {
     String vehicleType;
}
public class Car extends Vehicle {
     String modelType;
     public void showDetail() {
     vehicleType = "Car";
     modelType = "Sports";
     System.out.println(modelType +
" " + vehicleType);
     }
     public static void
main(String[] args) {
     Car car = new Car
     car.showDetail();
     }
}
```

EXAMPLE

```
class Animal {
     public void eat() {
     System.out.println("I can eat");
     }
     public void sleep() {
     System.out.println("I can sleep");
     }
}
```

```
class Dog extends Animal {
     public void bark() {
     System.out.println("I can bark");
     }
}
class Main {
     public static void main(String[] args)
{
     Dog dog1 = new Dog();
     dog1.eat();
     dog1.sleep();
     dog1.bark();
     }
}
```

RULES FOR INHERITANCE IN JAVA

- Multiple Inheritance is NOT permitted in Java
- Cyclic Inheritance is NOT permitted in Java
- Private members do NOT get inherited
- Constructors cannot be Inherited in Java
- In Java, we assign parent reference to child objects


THANK YOU