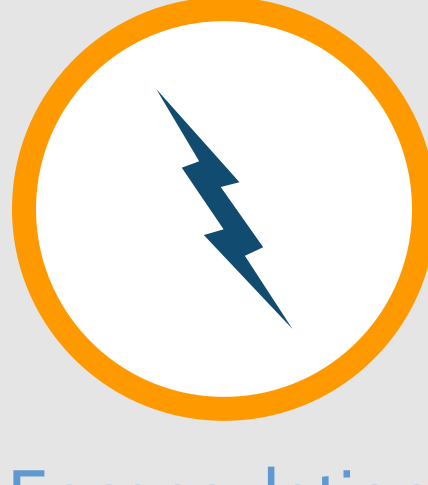


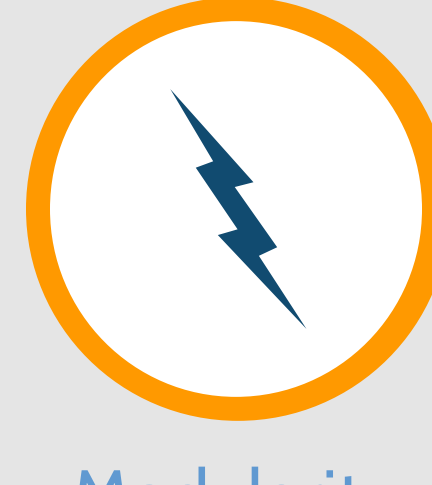
OBJECT ORIENTED PROGRAMING

KEY FEATURES OF OBJECT ORIENTED PROGRAMMING



Encapsulation

Groups of operations with datasets



Modularity

Process of solving the problem



Polymorphism

Using the same base class for a set of sub classes



Inheritance

Classes take the same properties as other classes

ENCAPSULATION



Refers to the grouping of data and the operations on that data in one package.



This is commonly used in the creation of classes as classes combine data with operation on the data.



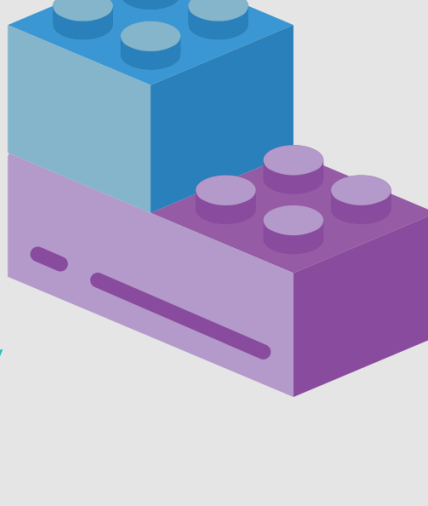
MODULARITY



Refers to the process of breaking up the main problem into smaller solvable problems.



These smaller parts are called modules and they are comprised of classes



POLYMORPHISM



Refers to different classes being able to be described with the same structure



Can change the interface of the base class to accommodate for the property's of the subclass



INHERITANCE



Refers to the class taking the properties of another class



Inheritance can be done with multiple classes and can be used with classes already using inheritance



ACCESS MODIFIERS

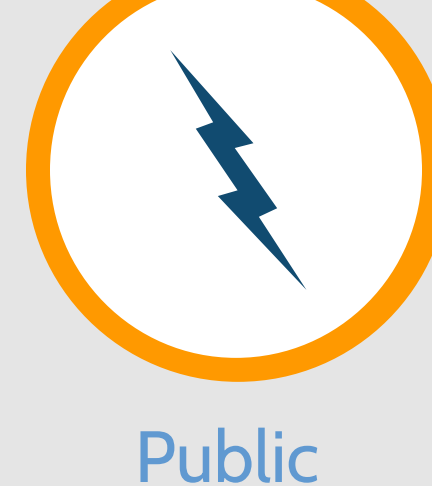


Restricts the availability of certain classes and variables to other classes or other programs entirely



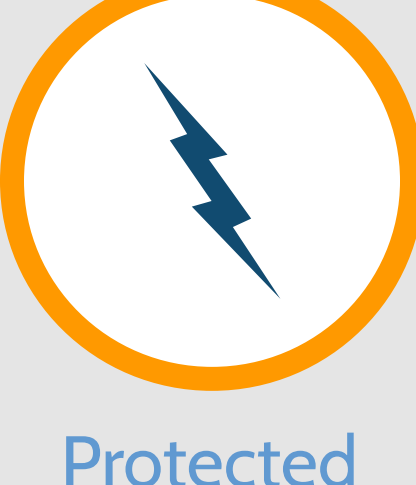
Default

If the class has no access modifier



Public

Available to all programs



Protected

Class can only be accessed by other subclasses



Private

Class cannot be accessed by any other class or program

Same Class

Same Package Subclass

Same Package Non-Subclass

Different Package Subclass

Different Package Non-Subclass

Public



Protected



Default



Private



PUBLIC



A class or variable with the modifier public can be accessed across any program



Widest scope out of all the modifiers



PROTECTED



Classes and variables with protected can be accessed by other classes in the same package



Only subclasses in other packages can access protected classes and variables in different packages



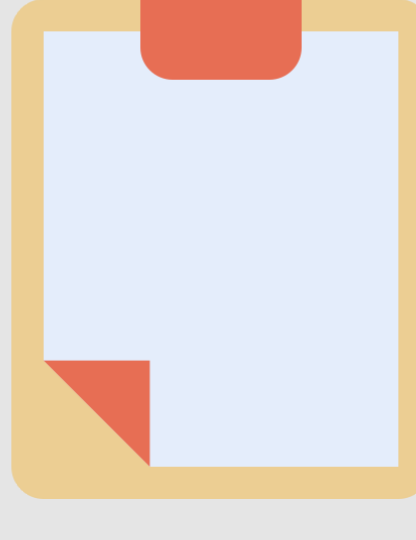
DEFAULT



Can only be stated with no access modifier



Can only be accessed by classes and subclasses in the same packag



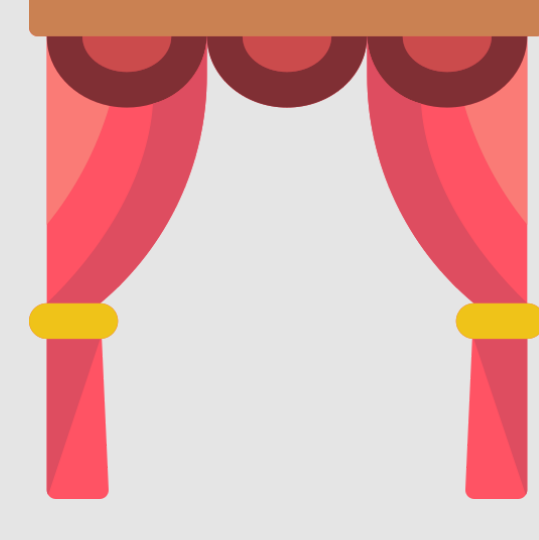
PRIVATE



Can only be accessed by the class it is stated in



Cannot be accessed by other classes in the same package



SHIFT FROM PP TO OOP



Derived from structured programming, calls for the creation of programs. Programs are comprised out of lines of code designed to do one function.



Based on the concept of objects. These objects carry sets of code and data that can interact with other objects. Most programming languages use a class based method of OOP.



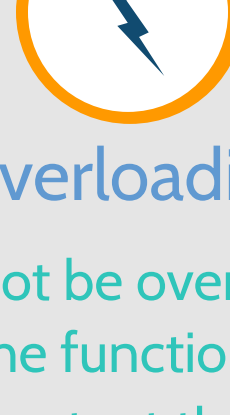
Different Approach

- PP uses a top down method as no access specifiers are present
- OOP uses a bottom up method as access specifiers are present



Adding Functions and Data

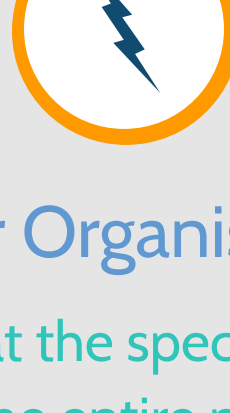
- PP has a significantly harder time changing function as it is less secure with no access specifiers present
- OOP does not have this shortfall as access modifiers are present



Overloading

- PP cannot be overloaded as in PP, the function is more important than the data
- OOP can be overloaded as OOP has it so the data is more important than the function

ADVANTAGES OF OOP OVER OTHER METHODS



Better Organisation

Can look at the specific object instead of the entire program like before



Problem Solving

Breaking the large complicated problem into smaller easier problems



Code Redundancy

Removes code redundancy using inheritance and polymorphism



Troubleshooting

Locating the error that is not letting the code to run



BETTER ORGANIZATION



Functions can be defined in small classes and can be referred back to if errors occur



Functions can be used an infinite amount of times if need be



PROBLEM SOLVING



Large complicated problems can use small classes to solve smaller problems within the problem



Can form a solution to the larger problem by solving the smaller problems



CODE REDUNDANCY



Repeating code can be avoided by using inheritance and polymorphism



Can organize similar classes using inheritance



TROUBLESHOOTING



Problems in the code are easily located as the issues reside in classes



Issues in classes reduces the amount of lines to sift through to find the error



BIBLIOGRAPHY

[Access Modifiers in Java](#)

[What is Encapsulation in OOP?](#)

[Modularity definition and information](#)

[OOP Concepts for Beginners: What is Polymorphism](#)

[OOP Concept for Beginners: What is Inheritance?](#)

[Differences between Procedural and Object Oriented Programming](#)

[4 Benefits of Object-Oriented Programming](#)

[Photos](#)

