|  |
| --- |
| **Inheritance** |
|  |  |
|  | Inheritance is a process of creating new classes from existing classes to achieve reusability. |
|  | The class which is being inherited is known as 'Super class' and the class which inherits the other class is known as 'Sub class' |
|  | this allows all subclasses to inherit all properties of their parent classes. |
|  |  |
|  | Inheritance represents the **IS-A** relationship |
|  |  |
|  | There are different types of inheritance: |
|  | 1.Single Inheritance : only one super class |
|  | 2.Multiple Inheritance : many super classes |
|  | 3.Hierarchial Inheritance : one super class and and many subclasses |
|  | 4.Multi-level Inheritnace : class derived from a derived class |
|  | 5.Hybrid Inheritance : more than two types of forms(Multiple + MultiLevel) |
|  |  |
|  |  |
|  |  |
|  | **Benfits of Inheritance:**  we may have many benfits : |
|  | **Software ReUse :**  When we inherit from class we no need to write the behaviour of super class. |
|  | **Code Sharing :**  When two classes are inheriting same super class then the super class code can be shared in both the subclasses. |
|  | **Improved Reliability :**  When we use inheritance repeatedly, we will have lesser bugs in that code because we are going through it every time we use inheritance |
|  | **Consistency :**  when two classes inherit same super class then we can assure that same behaviour in all cases. |
|  | **Rapid Prototyping :**  When we are developing a system with reusable components then we will get a chance to focus on new requirements. |
|  | **Polymorphis :**  Inheritance allows subclasses to create high-level reusable components that can be specific need fit different applications by changes in their low level parts. |
|  | **Infomation Hiding :**  while implementing inheritance we no need to understand the super class. We just need to understand the nature of the component and its interface for accessing. |
|  |  |
|  | **Costs of inheritance :** |
|  | **Execution Speed :**  The execution time for specialized methods in sub-classes is more faster than methods inherited from super class. |
|  | **Program size :**  The use of any software library imposes a size penalty interms of speed and memory. |
|  | **Program Complexity :**  Over use of inheritance will make code complexity.  Java doesn’t Support multiple inheritance  Refer below link for getting to know about multiple inheritance because of diamond problem.  <https://www.digitalocean.com/community/tutorials/multiple-inheritance-in-java>  Java supports multiple inheritance for interfaces.  Problem while implementing multiple inheritance:  When a subclass is extending multiple super classes  If the super classes are also extending same parent class then the grandchildren class doesn’t know which method to invoke. Then compiler will throw ambiguity.  It will be okay with interfaces because we will always invoke same implemented class. |
|  |  |