# Task1:

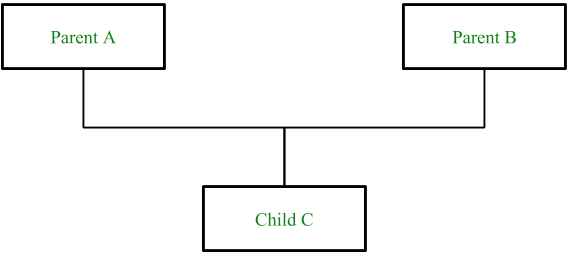
* Explain the advantage and disadvantage of multiple inheritance. Write a PHP program to demonstrate that.

1. Advantage

* Inheritance promotes reusability. When a class inherits or derives another class, it can access all the functionality of inherited class.
* Reusability enhanced reliability. The base class code will be already tested and debugged.
* As the existing code is reused, it leads to less development and maintenance costs.
* Inheritance makes the sub classes follow a standard interface.
* Inheritance helps to reduce code redundancy and supports code extensibility.
* Inheritance facilitates creation of class libraries.

1. Disadvantage

* Inherited functions work slower than normal function as there is indirection.
* Improper use of inheritance may lead to wrong solutions.
* Often, data members in the base class are left unused which may lead to memory wastage.
* Inheritance increases the coupling between base class and derived class. A change in base class will affect all the child classes.



Syntax**:**

|  |
| --- |
| class child\_class\_name extends parent\_class\_name {  use trait\_name;  ...  ...  child\_class functions  } |

Example code:

|  |  |
| --- | --- |
| Input | Output |
| <?php  class Foo {     public function printItem($string)     {         echo 'Foo: ' . $string . PHP\_EOL;     }          public function printPHP()     {         echo 'PHP is great.' . PHP\_EOL;     } }  class Bar extends Foo {     public function printItem($string)     {         echo 'Bar: ' . $string . PHP\_EOL;     } }  $foo = new Foo(); $bar = new Bar(); $foo->printItem('baz'); // Output: 'Foo: baz' $foo->printPHP();       // Output: 'PHP is great'  $bar->printItem('baz'); // Output: 'Bar: baz' $bar->printPHP();       // Output: 'PHP is great'  ?> | Foo: baz  PHP is great.  Bar: baz  PHP is great. |