- Dependency is normally created when you receive a reference to a class as part of a particular operation/method. Dependency indicates that you may invoke one of the APIs of the received class reference and any modification to that class may break your class as well. Dependency is represented by a dashed arrow starting from the dependent class to its dependency.

- Class Die { public void Roll() { ... } }
  class Player
  {
  public void TakeTurn(Die die) /*O'm dependent on Die and it's Roll method to do my work*/
  { die.Roll(); ... }
  }

- Dependency is a weaker form of bond: one class depends on another because it uses it at some point in time.

- One class depends on another if the independent class is a parameter variable or local variable of a method of the dependent class. This is different from an association, where an attribute of the dependent class is an instance of the independent class. Sometimes the relationship between two classes is very weak. They are not implemented with member variables at all. Rather they might be implemented as member function arguments.

Class diagram showing dependency between "Car" class and "Wheel" class (An even clearer example would be "Car depends on Wheel", because Car already *aggregates* (and not just *uses*) Wheel)