OOP Test 8.1 definitions

Nguyen Manh Duc ID: 103792724

# Abstraction

The idea of abstraction in programming holds that extra information should be "hidden" from the user. We abstract away the process for locating a book or game in the library in the test program. Yes, the function's function is specified (HasResource), but the implementation behind it is concealed. In essence, we try to write the program in a "descriptive" manner, much like to how math or English are written. Most individuals realize of what does, but few could compute a nontrivial root.

The functional implementation, state data in objects, and other features are all included in these concealed implementation details. To finish a function effectively, we should hide everything that isn't necessary. For example, a math function only requires the input number, and a hasResource function just need the name of the resource.

# Polymorphism

Polymorphism in object-oriented programming refers to the ability of a type of data or function to take on many different forms (definitions). In 8.1, LibraryResource, Game, and Book are examples of polymorphism. If you want to use LibraryResource, you can use games or books instead.It is guaranteed that the LibraryResource can perform all the roles it can take, so it is safe for the programmer to call a method defined as follows: LibraryResource is because both Game and Book inherit these methods.

There are many other forms of polymorphism in OOP.For example, different method implementations with the same name maintain the same "result" functionality despite differences elsewhere. There is a lot of discussion surrounding the potential benefits of meditation, but there is little evidence to support the claims. There is much discussion surrounding the potential benefits of meditation, but little evidence to support the claims.