**CHAPTER 5**

ENCAPSULATION, ABSTRACTION, POLYMORPHISM

**Data Abstraction**

\*\* Data abstraction refers to, providing only essential information to the outside world and hiding their background details, i.e., to represent the needed information in program without presenting details.

\*\* Data abstraction is a programming and design technique that relies on the separation of interface and implementation.

\*\* Real life example of a TV, which you can turn on and off, change the channel, adjust the volume, and add external components such as speakers, VCRs, and DVD players, BUT you do not know its internal details, that is, you do not know how it receives signals over the air or through a cable, how it translates them, and finally displays them on the screen.

**Benefits of Data Abstraction**

\*\* Class internals are protected from inadvertent (unintended) user-level errors, which might corrupt the state of the object.

\*\* Provides only the essential information to the outside world and hides the background details, so the interface remain secure.

\*\* The class implementation may evolve over time in response to changing requirements or bug reports without requiring change in user-level code.

\*\* Separates the interface from implementation, so the code can be modified afterwards without changing the interface.

**Abstract Class (Interface)**

\*\* An interface describes the behavior or capabilities of a C++ class without committing to an implementation of that class.

\*\* The C++ interfaces are implemented using abstract classes and these abstract classes should not be confused with data abstraction which is a concept of keeping implementation details separate from associated data.

***[ Note by -Jannatul Ferdous Umama(Bristy)]***

\*\*The purpose of an abstract class is to provide an appropriate base class from which other classes can inherit.

\*\* Abstract classes cannot be used to instantiate objects and serves only as an interface. Attempting to instantiate an object of an abstract class causes a compilation error.

\*\* Thus, if a subclass of an ABC needs to be instantiated, it has to implement each of the virtual functions, which means that it supports the interface declared by the ABC.

\*\* Failure to override a pure virtual function in a derived class, then attempting to instantiate objects of that class, is a compilation error.

\*\*Classes that can be used to instantiate objects are called concrete classes.

**Polymorphism**

\*\* The word polymorphism means having many forms.

\*\* Typically, polymorphism occurs when there is a hierarchy of classes and they are related by inheritance.

\*\* C++ polymorphism means that a call to a member function will cause a different function to be executed depending on the type of object that invokes the function.

\*\* Function Overloading is also called Compile time Polymorphism.

***[ Note by -Jannatul Ferdous Umama(Bristy)]***