**Task 1: What is the difference between interface class, abstract class, and abstract method?**

* An abstract class is a class that cannot be instantiated on its own and is intended to be a base class for other classes. It may contain abstract methods, which are methods without implementation, and it may also contain concrete methods, which are methods with implementation.
* **Abstract Methods**: These methods are declared but not implemented in the abstract class. Subclasses that derive from the abstract class must implement these methods.
* **Concrete Methods**: An abstract class can also have methods with full implementation that can be inherited by subclasses.
* **Instance Variables**: An abstract class can have instance variables (fields) and constructors.
* **Inheritance**: A class can inherit from only one abstract class (single inheritance).
* An interface is a reference type in object-oriented programming that is used to specify a set of methods that a class must implement. Unlike an abstract class, an interface typically cannot have any concrete methods or instance variables; it only defines the method signatures.
* Abstract Methods: All methods in an interface are abstract, meaning they have no implementation.
* No Instance Variables: Interfaces cannot have fields (instance variables) or constructors.
* Multiple Inheritance: A class can implement multiple interfaces, which allows for a form of multiple inheritance.