=> interface :-

100% abstraction :-

==> Through interface we can achieve 100% abstraction

==> interface can be used to specify our requirement which to be implemented

==> All the methods in an interface are public and abstract irrespective wether we specify or not

==> It's compulsory for an implementing class to override all methods on an interface if implementing class is not overriding all methods of an interface then it must be declared as abstract

==> One interface can have any number of implementing classes

==> We cannot create object/instance of an interface

==> We can create reference of an interface to achieve polymorphism

==> A class will implement an interface using 'implements' keyword

==> A class can implement multiple interfaces

==> A class can extends another class also implement one or more interfaces at a time , in this first it must extends a class then implements an interface

==> An interface cannot implements another interface

==> An interface can extend another interface

==> In an interface we can have variable however all the variables in an interface are public static final by default;

==> We cannot have constructor in an interface

==> From Java 1.8 or Java 8 in an interface we can have method with body/implementation however that method must be implemented using default keyword.

==> default methods of an interface will get inherited to all implementing classes however its not compulsory for an implementing classes to override default methods of an interface, if need arises we can override also

==>We can also have static methods in an interface however static methods of an interface must not be abstract it must have implementation. And static method of an interface will not get inherited in implementing classes.

==> An implementing class can have specialized methods also however using interface type refrence we cannot access directly but can be done through down casting.