Difference between Interface and Abstract class?

Abstraction concept:

Both Interface and Abstract class are used to achieve abstraction concept.

Abstract class can have both abstract methods and non-abstract methods

Abstraction means:

to hide some implementation logic and that particular logic will be defined in child class.

Since abstratction class have both abstract methods and non-abstract methods,it is called as partial -abstratction mechanism.

Abstract class:

Keyword: Abstract shd be used.

Rules:

For a class to be declared as abstract, there shd be atleast one method which is abstract in nature.

Ot can also have n no. of non-abstract methods also.

Abstract method?

Only prototype.only method declaration,no method body.

e.g. public abstract void loan();

// no method body here.

🡪cannot create object of abstract class

🡪we can define any variable over here (like final or non final var or static or non static )

dynamic polymorphism cannot be achieved.

Interface:

Inside interface,we can achieve 100% abstratction.

That is,inside interface,all methods are abstract by default.

No need to use abstract keyword,bcz we define only abstract methods in interface always.

In interface,we can define only final and static variables are defined here.

E.g:

Class 1:Car class -parent interface

**public** **interface** Car {

**public** **void** start();

**public** **void** stop();

**public** **void** refuel();

}

🡪no implementation logic in interface level,that’s y it is 100% abstraction.

cannot create object of interface .

🡪dynamic polymorphism can be achieved here.i.e. child class obj referred by parent interface ref variable.

Car c= new BMW();

Class 2: BMW implements parent interface(do not use extend here,we have to use implements to use interface methods)

**public** **class** BMW **implements** Car {

@Override

**public** **void** start() {

System.***out***.println("BMW---start");

}

@Override

**public** **void** stop() {

System.***out***.println("BMW---stop");

}

@Override

**public** **void** refuel() {

System.***out***.println("BMW---refuel");

}

}

When to use Abstract and interface?

Whenever we have to achieve,partial abstraction

e.g.some methods can be common for all the classes.in that case,we can use abstract class.Performance wise,abstract class is faster that interface.

When we want to hide complete business logic,we have to use interface.