Difference Between Thread Class and Runnable Interface in Java

A thread can be defined in two ways. First, by **extending a Thread class** that has already implemented a Runnable interface. Second, by directly **implementing a Runnable interface**. When you define a thread by extending Thread class you have to override the run() method in Thread class. When you define a thread implementing a Runnable interface you have to implement the only run() method of Runnable interface.

The basic difference between Thread and Runnable is that each thread defined by extending Thread class creates a unique object and get associated with that object. On the other hand, each thread defined by implementing Runnable interface shares the same object.

Comparison Chart

BASIS FOR COMPARISON	THREAD	RUNNABLE
Basic	Each thread creates a unique object and gets associated with it.	Multiple threads share the same objects.
Memory	As each thread create a unique object, more memory required.	As multiple threads share the same object less memory is used.
Extending	In Java, multiple inheritance not allowed hence, after a class extends Thread class, it can not extend any other class.	If a class define thread implementing the Runnable interface it has a chance of extending one class.
Use	A user must extend thread class only if it wants to override the other methods in Thread class.	If you only want to specialize run method then implementing Runnable is a better option.
Coupling	Extending Thread class introduces tight coupling as the class contains code of Thread class and also the job assigned to the thread	Implementing Runnable interface introduces loose coupling as the code of Thread is separate form the job of Threads.