Difference Between Runnable and Thread

www.differencebetween.com

Key Difference - Runnable vs Thread

A program in execution is known as a process. The process can be divided into multiple subprocesses. For example, Microsoft Word is a process. At the same time, it checks for the spelling mistake and grammar. That is a subprocess. These subprocesses are known as threads. Multithreading is the process of executing multiple threads simultaneously. Threaded applications can be built using different programming languages. Runnable and Thread are associated with Java programming. There are two methods in Java to create a thread by implementing a Runnable interface or extending the Thread class. **When implementing Runnable, many threads can share the same thread object while in Extending Thread class, each thread has a unique object associated with it.** That is the **key difference** between Runnable and Thread.

What is Runnable?

A thread goes through some states. The "new" is the beginning of the thread life cycle. After the start() method calls on a new thread, it becomes runnable. If the thread scheduler selects the thread, it transits to the running state. The thread is waiting for the state if that thread is waiting for another thread to perform a task. After the thread completes the work, it goes to the termination state.

A thread can be implemented using the Runnable interface. Refer the below program.

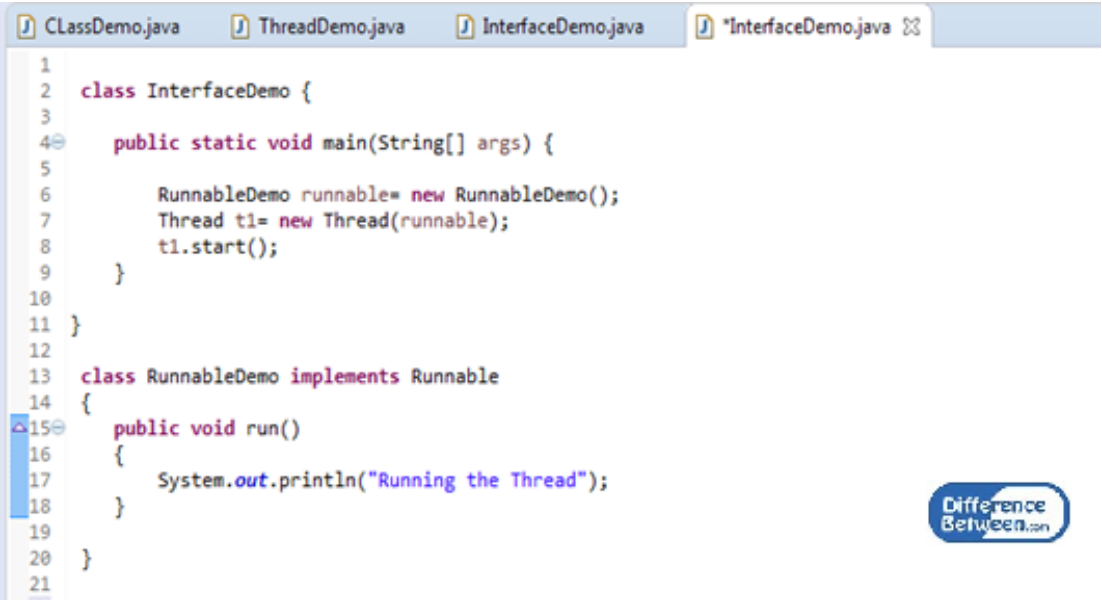

Figure 01: Java Program to create a thread using Runnable interface

According to the above program, the class Runnable Demo implements the Runnable interface. The run() method is inside the class that implements Runnable interface. It is the entry point for the thread. The logic is in the run() method. In the main program, a thread is created by defining an object that instantiated from Runnable Demo class. It is t1. The start() method is called using t1.

Refer the below program.

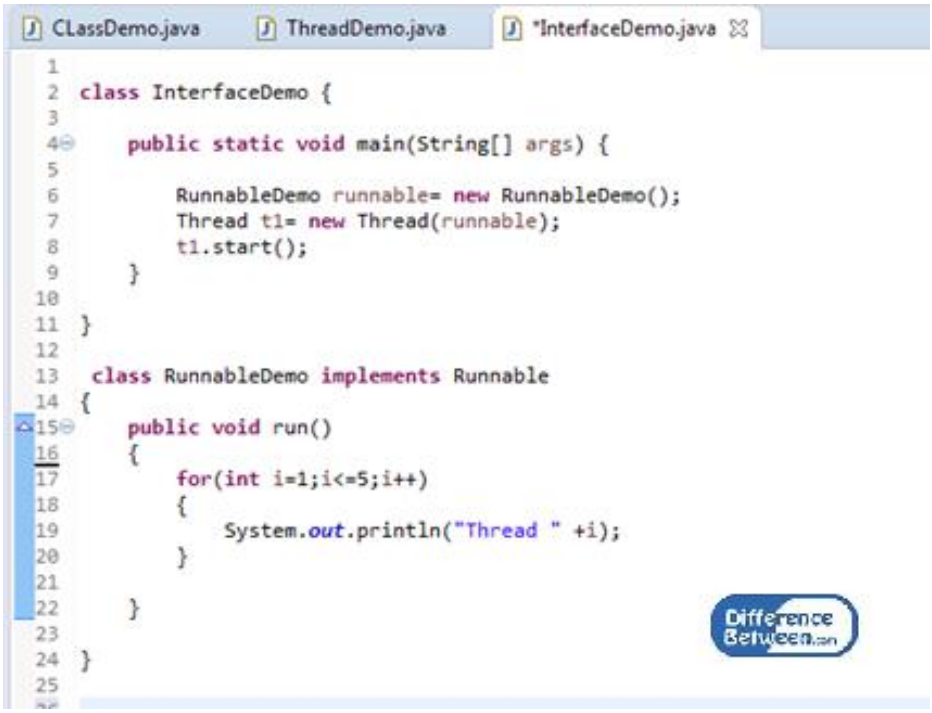

Figure 02: Java program to create a thread to execute a loop, by implementing the Runnable interface

According to the above example, the class Runnable Demo implements Runnable interface. The logic to execute using the thread is written in run() method. In the main program, a thread is created by defining an object that instantiated from Runnable Demo class. It is t1. Then, the start() method is called using t1.

What is Thread?

The other method of creating a thread is by extending the Thread class. It consists of three steps. First is to declare the class as extending the Thread class. Afterwards, the run() method should be written. It has the sequence of steps the thread should execute. Finally, the thread object is created, and the start() method is called to initiate the execution of the thread. Refer the below program.

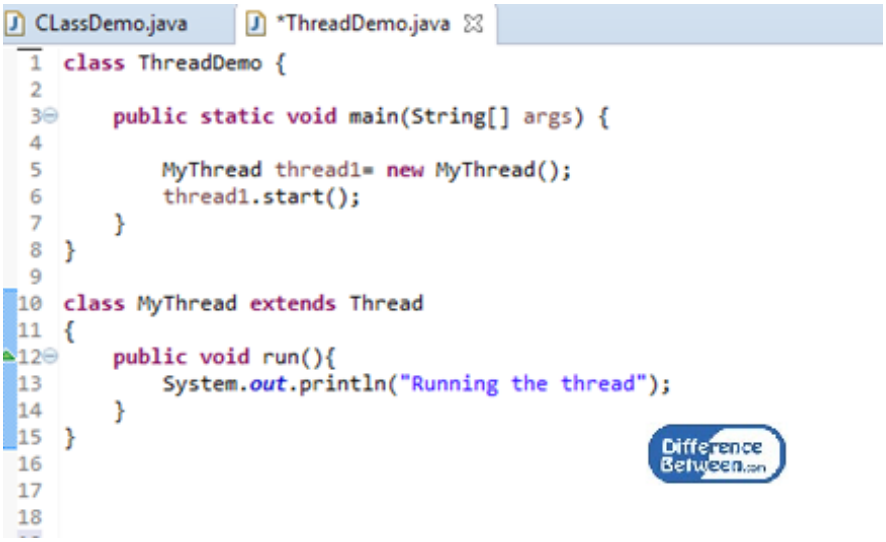

Figure 03: Java program that extends Thread class

According to the above program, the MyThread class extends Thread class. It overrides the run method. The run() method contains the logic to be executed by the thread. It is the entry point to the thread. Then the thread object is created. It is thread1. The thread is started using the start() method. It will execute a call to run() method.

An example program of two classes extending the Thread class is as follows.

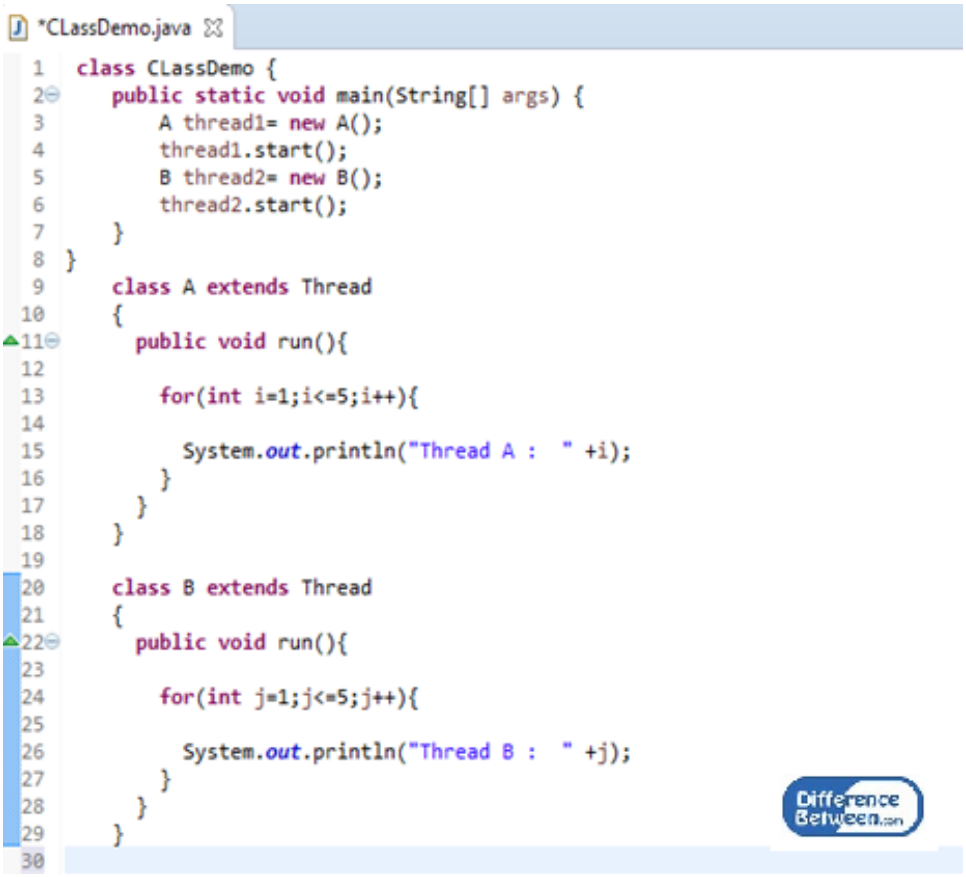

Figure 04: Java program with two classes that extend Thread class

According to the above program, both class A and B are extending Thread class. Both classes have their implementation of the run() method. The main thread is which executes the main() method. Before main thread dies, it creates and starts thread1 and thread2. By the time the main thread reached the end of the main method, three threads are running in parallel. There is no specific order in which the threads give output. Once the thread is started it is hard to decide the order they will execute. They run independently.

What are the Similarities Between Runnable and Thread?

- Both are using to create a thread in Java.

What is the Difference Between Runnable and Thread?

Runnable vs Thread	
Runnable is an interface in Java to create a	The thread is a class in Java to create a thread

thread that allows many threads to share the same thread object.	where each thread has a unique object associated with it.
Memory	
In Runnable, multiple threads share the same object, so require less memory.	In Thread class, each thread creates a unique object, therefore requires more memory.
Extending Ability	
After implementing Runnable interface, it can extend a class.	Multiple inheritances are not supported in Java. After extending Thread class, it cannot extend any other class.
Code Maintainability	
Runnable interface makes code more maintainable.	In Thread class, maintaining is time-consuming.

Summary - Runnable vs Thread

A process is divided into multiple sub-processes to perform multiple tasks at the same time. These subprocesses are known as threads. Instantiating a thread can be done by implementing the Runnable interface or by extending the Thread Class. It is easy to extend the Thread class, but it is not a better Object-Oriented Programming practice. When implementing Runnable, many threads can share the same thread object while in extending Thread class each thread has a unique object associated with it. That is the key difference between Runnable and Thread. In Thread class multiple object creation can consume more memory.

Reference:

1.tutorialspoint.com. “Java Multithreading.” The Point, Available here
2.Pramodbablad. “Extends Thread Vs Implements Runnable In Java.” Java Concept Of The Day, 11 Nov. 2016. Available here
3.Ways to create a Thread in Java Multithreading | Core Java Tutorial | Studytonight. Available here

How to Cite this Article?

APA: Difference Between Runnable and Thread.(2018 January 23). Retrieved (date), from http://differencebetween.com/difference-between-runnable-and-vs-thread/

MLA: "Difference Between Runnable and Thread" Difference Between.Com. 23 January 2018. Web.