Multithreading

threads can be executed simultaneously in multiprocessor systems, In single-processor systems, as shown in Figure 29.1(b), the multiple threads share CPU

time known as *time sharing*, and the operating system is responsible for scheduling and allocating

resources to them

When your program executes as an application, the Java interpreter starts a thread for the

main method. When your program executes as an applet, the Web browser starts a thread to

run the applet.

You can create additional threads to run concurrent tasks in the program. In

Java, each task is an instance of the Runnable interface, also called a runnable object. A

thread is essentially an object that facilitates the execution of a task.

Tasks are objects. To create tasks, you have to first define a class for tasks. A task class must

implement the Runnable interface. The Runnable interface is rather simple. All it contains

is the run method. You need to implement this method to tell the system how your thread is

going to run.

A task must be executed in a thread.