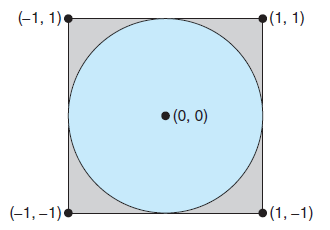
**Operating Systems**

**Exercise 2**

**Monte Carlo technique for calculating pi (multithreaded version)**

An interesting way of calculating PI is to use a technique known as ***Monte Carlo,*** which involves randomization. This technique works as follows: suppose you have a circle inscribed within a square, as shown in the following figure.



Assume that the radius of this circle is 1. First, generate a series of random points as simple (*x, y*) coordinates. These points must fall within the Cartesian coordinates that bound the square. Of the total number of random points that are generated, some will occur within the circle. Next, estimate PI by performing the following calculation:

PI = 4× *(number of points in circle)* / *(total number of points)*

Write a multithreaded version of this algorithm that creates a separate thread to generate a number of random points. The thread will count the number of points that occur within the circle and store that result in a global variable. When this thread has exited, the parent thread will calculate and output the estimated value of PI. It is worth experimenting with the number of random points generated. As a general rule, the

greater the number of points, the closer the approximation to PI.