24-780 Engineering Computation Problem Set 12

You need to create a ZIP file (It may appear as a compressed folder in Windows) and submit the ZIP file via the 24-780 Canvas course. The file name of the ZIP file must be:

PS12-YourAndrewID.zip

For example, if your Andrew account is *hummingbird@andrew.cmu.edu*, the file name must be:

PS12-hummingbird.zip

If your ZIP file does not comply with this naming rule, you will automatically lose 5% credit from this assignment. If we are not able to identify who submitted the file, you will lose another 5% credit. If we finally are not able to connect you and the submitted ZIP file, you will receive 0 point for this assignment. Therefore, please make sure you strictly adhere to this naming rule before submitting a file.

The ZIP file needs to be submitted to the 24-780 Canvas course. If you find a mistake in the previous submission, you can re-submit the ZIP file with no penalty as long as it is before the submission deadline.

Notice that the grade will be given to the final submission only. If you submit multiple files, the earlier version will be discarded. Therefore, if you re-submit a ZIP file, the ZIP file MUST include all the required files. Also, if your final version is submitted after the submission deadline, late-submission policy will be applied no matter how early your earlier version was submitted.

Make sure you upload your Zip file to the correct location. If you did not upload your assignment to the correct location, you will lose 5%.

The ZIP file needs to include:

- C++ source file of your program (ps12.cpp)

Submission Due: Please see Canvas.

START EARLY!

Unless you are already a good programmer, there is no way to finish the assignment overnight.

PS12 Accelerate the calculation of π with Parallel Processing [ps12.cpp] (100 points)

The following program estimates the value of π by sampling points over a rectangular lattice. Similar idea as the Monte-Carlo, but in this problem we don't use random sample points.

Modify the program so that it calculates the value in parallel using std::thread. Your program should use eight threads.

Hint: For thread K (0<=K<4), the range of x should be:

```
res*K/4<=x<res*(K+1)/4
```

The program must be saved as ps12.cpp and submitted to the Canvas in the zip file.

```
#include <stdio.h>
#include <time.h>

int main(void)
{
    const int res=40000;

    auto t0=time(nullptr);

    int M=0;
    const int N=res*res;
    for(int x=0; x<res; ++x)
    {
        for(int y=0; y<res; ++y)
        {
            const double xx=(double)x/(double)res;
            const double yy=(double)y/(double)res;

            if(xx*xx+yy*yy<=1.0)
            {
                M++;
            }
        }
    }

    const double pi=2.0*2.0*(double)M/(double)N;
    printf("Estimated PI=%lf\n",pi);
    printf("Time Elapsed=%d\n",(int)(time(nullptr)-t0));

    return 0;
}
```