Object-Oriented Programming Language

Homework Assignment No. 2

Due 09:00 pm, Wednesday September 28, 2016
Late submission within 24 hours: score*0.9;
Late submission before post of solution: score*0.8 (the solution will usually be posted
within a week); no late submission after the post of solution)

(Total 80%)

1. (20%) For the sum $S(N) = 1 + \dfrac{1}{2} + \dfrac{1}{3} + \dfrac{1}{4} + ... + \dfrac{1}{N}$, write a C++ program to calculate the smallest integer N such that $S(N) > M$ where M is a user input. Below is a typical run:


```
C:\WINDOWS\system32\cmd.exe
Please input M : 3.2
1/1+1/2+1/3+1/4+1/5+1/6+1/7+1/8+1/9+1/10+1/11+1/12+1/13+1/14=3.25156

The smallest integer N=14
請按任意鍵繼續 . . .
```
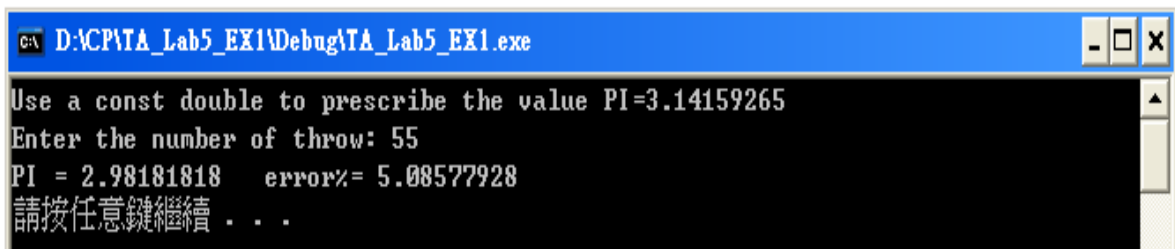
2. (30%) 蒙地卡羅法(Monte Carlo Method)求圓周率
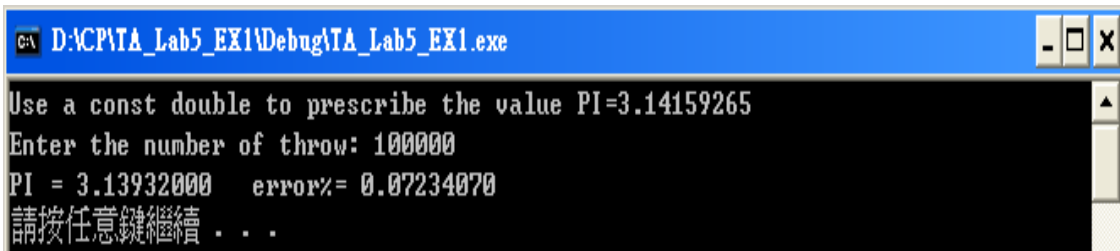蒙地卡羅法(Monte Carlo Method)求圓周率的原理示意圖如下。正方形邊長為 1 單位長，面積為 1 平方單位；黃色扇形面積等於半徑為 1 單位長的 1/4 圓，面積為 PI/4。在正方形內均勻隨機丟石頭 (或隨機擲飛鏢)

落在扇型內的機率 ＝ 扇型面積 ÷ 正方形面積 ＝PI/4。
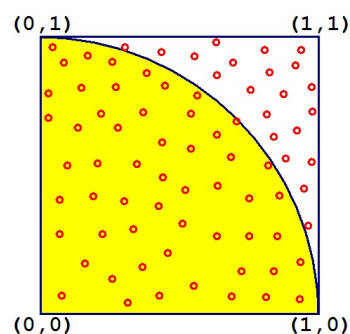
Follow the note given above and write a Monte Carlo method to compute PI. You should ask the user to input the desirable number of random numbers and observe as the number gets bigger, your result should be closer to PI. Use a const double to prescribe the value PI=3.14159265 and report the error in % obtained from the Monte Carlo method. Below are sample runs:

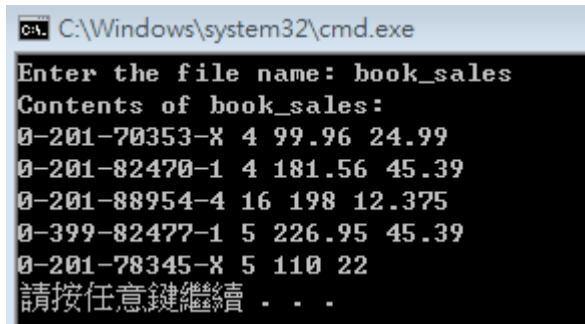3. **(30%)** Download the folder `HW2_dist.zip` from the course website and unzip the folder. You will find `Sales_item.h`, a data file `book_sales` that contains a few book records and a supplementary note on input file stream from my Computer Programming course. To ease your life for now, you can safely assume the records for each ISBN is grouped together.

   Write a C++ program to read the transactions from the data file `book_sales` and output on a screen that shows, for each book, the total number of copies sold, the total revenue, and the average sales price. (**hint**: check (1) the supplementary note given in the `HW2_dist.zip` and (2) C++ primer Section 8.4 if you are not familiar with C++ input file stream)

   Your program should ask users to enter the data file name to be processed. It should complain if it cannot find the data file. Two sample outputs look like the following: the first shows the sample that cannot find the file and terminate the run and the second sample does the regular processing successfully.

Object-Oriented Programming Language

HW Grading Policy:

1. You should consider about exception handling, e.g. error input, file opening fail, etc.
   請注意所有例外狀況的處理，例如：錯誤的符號字串輸入、檔案開啟失敗等。

2. The coding style includes your output format.
   輸出資料的格式將納入格式評分。

3. If your code is not compilable, your score in this problem is zero (including coding style).
   若程式無法編譯，則該題以零分計算。(包含格式分數)

4. Your program will be tested with other data which is not the same as provided samples.
   除了題目所提供的範例測試資料以外，作業程式碼將以額外的測試資料進行測試。

5. If tricky situations occur, the grade depends on Prof. Chen or TA's judgment.
   假如有特殊情況發生，則依據陳俊杉教授以及助教們的判斷給分。

- Coding Style (20%): 編碼格式分數
    1. format
       整體形式與輸出資料的格式
    2. comments
       註解
    3. readability
       可讀性
    4. variables naming
       變數命名方式
    5. typesetting
       型別設定

- Functionality (80%): 功能性分數
    1. run-time performance:
       執行時的表現
        1) samples not passed -> x
           範例測資錯誤 => 此部分零分
        2) samples passed but some tests failed -> partial
           範例測資通過但是部分測資失敗 => 部份給分

3) samples and tests all passed
範例測資與所有測資通過 => 此部分滿分

3. excellent method++
綜合以上，又以能展現解決問題的巧思尤佳。