24-780 Engineering Computation Problem Set 07

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. The file name of the ZIP file must be:

PS07-YourAndrewID.zip

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

PS07-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. 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 (ps7-1.cpp, ps7-2.cpp)

Please do not include other files generated by an IDE.

Submission Please see Canvas

START EARLY!

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

PS7-1 Rewrite PS2-2 using a C++ class

(1) **Download ProblemSet2 solution from Canvas**
(2) **Rename ps2-3-solution.cpp to ps7-1.cpp**
(3) **Do not use scanf. Use fgets and atoi instead.**
(4) **Add the following class definition:**

```
class FlashCard
{
public:
    int a,b;
    void PrintCard(void);
    int CorrectAnswer(void);
};
```

(5) **Write PrintCard and CoorectAnswer member functions. PrintCard member function must prompt the user to enter the solution, and CorrectAnswer function must return the correct answer for this card.**
(6) **Replace:**

```
int card[144],nCard;
```

in the main function with:

```
FlashCard card[144];
int nCard;
```

(7) **Then, re-write the rest of the code accordingly. For full credit, you need to:**
    a. **Change SwapInt function to SwapFlashCard function, and make it swap two FlashCard objects.**
    b. **Shuffle card function must take an array of FlashCard objects rather than an array of integers.**
    c. **Actually use PrintCard and CorrectAnswer member functions**

PS7-2 Rewrite PS4-2 using C++ classes

(1) **Download ProblemSet4 solution from Canvas**

(2) **Rename ps4-2-solution.cpp to ps7-2.cpp**

(3) **Add the following class definitions:**

```
class CannonBall
{
public:
    int state;
    double x,y,vx,vy;
    void Draw(void);
};
class Target
{
public:
    double x,y,w,h;
    void Draw(void);
    int IsHitByBall(CannonBall &ball);
};
class Obstacle
{
public:
    int state;
    double x,y,w,h;
    void Draw(void);
    int IsHitByBall(CannonBall &ball);
};
```

(4) **Replace:**

```
int canonState;
double canonX,canonY,canonVx,canonVy;
```

with:

```
CannonBall cannonBall;
```

and, replace:

```
double targetX,targetY,targetW,targetH;
```

with:

```
Target target;
```

and, replace:

```
int obstacleState[nObstacle];
double obstacleX[nObstacle],obstacleY[nObstacle];
double obstacleW[nObstacle],obstacleH[nObstacle];
```

with:

```
Obstacle obstacle[nObstacle];
```

(5) **DrawCannon and DrawTarget functions must be replaced with CannonBall::Draw and Target::Draw functions respectively.**

(6) **Obstacle::Draw function must use DrawRect function to draw an obstacle self.**

(7) **DrawObstacle function must take the number of obstacles and an array of Obstacle objects, and use Obstacle::Draw function to draw an obstacle.**

(8) **Target::IsHitByBall and Obstacle::IsHitByBall functions both must return 1 if the target or the obstacle is hit by the ball located at (ballX,ballY), or 0 otherwise. (You can call CheckHitTarget**

function from inside these two member functions to make it easier, or you don't have to if it is easier for you.)

(9) Then, re-write the rest of the code accordingly. For full credit, you need to:

- **a. Change GenerateObstacle function so that it takes the number of obstacles and an array of Obstacle objects.**
- **b. Actually use Obstacle::Draw, Obstacle::IsHitByBall, CannonBall::Draw, Target::Draw, and Target::IsHitByBall functions.**

For example,

```
if(obstacleState[i]==1 &&
   CheckHitTarget(canonX,canonY,obstacleX[i],obstacleY[i],obstacleW[i],obstacleH[i])==1)
```

Should be replaced with:

```
if(obstacle[i].state==1 && obstacle[i].IsHitByBall(cannonBall)==1)
```

Hint: It is not a good idea to make all changes at once. Identify the smallest change you can make that still keeps the program functioning. Make the change and test. And then move on to the next. Repeat these steps until your program complies with all the requirements.

Common confusion: In the past, I have seen many mistakes that is making a function like:

```
CannonBall::Draw(double x,double y);
```

And calling it as:

```
cannonBall.Draw(cannonBall.x, cannonBall.y);
```

However, the cannonBall object knows its own position, and therefore Draw function does not need to take its own location as input parameters. Points will be taken off if you make this kind of mistakes.

You can think of more member functions. Also you can make the artillery as a class. All those changes will make the program cleaner. However, for this assignment, you will get full credit if your program complies with the above requirements.