24-780 Engineering Computation Problem Set 11

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:

PS11-YourAndrewID.zip

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

PS11-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 (ps11-1.cpp)
- C++ source file of your program (ps11-2.cpp)

Submission Due: Please see Canvas

START EARLY!

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

PS11-1 Using template (50 points) (ps11-1.cpp)

Read the following code (also available in the Canvas). It includes three overloaded functions called Summation. Replace the three functions with a single template function called Summation.

Also use const qualifier in appropriate place(s).

Save your program as ps11-1.cpp and include in the Zip file.

```
#include <stdio.h>

int Summation(int nElem,int elem[])
{
    int sum=0;
    for(int i=0; i<nElem; ++i)
    {
        sum+=elem[i];
    }
    return sum;
}

double Summation(int nElem,double elem[])
{
    double sum=0;
    for(int i=0; i<nElem; ++i)
    {
        sum+=elem[i];
    }
    return sum;
}

float Summation(int nElem,float elem[])
{
    float sum=0;
    for(int i=0; i<nElem; ++i)
    {
        sum+=elem[i];
    }
    return sum;
}

int main(void)
{
    int intArray[5]={1,2,4,8,16};
    double doubleArray[5]={8.1,9.1,10.1,11.1,12.1};
    float floatArray[5]={1.5f,2.5f,3.5f,4.5f,5.5f};

    printf("Summation of 1,2,4,8,16=%d\n",Summation(5,intArray));
    printf("Summation of 8.1,9.1,10.1,11.1,12.1=%lf\n",Summation(5,doubleArray));
    printf("Summation of 1.5,2.5,3.5,4.5,5.5=%f\n",Summation(5,floatArray));

    return 0;
}
```

PS11-2 Using virtual functions (50 points) (ps11-2.cpp)

Read the following program (also available in the Canvas). Pay attention to the main function and see how Move and Draw member functions are used. The goal of this program is to draw one blue square that can be moved by arrow keys, and one green diamond that chases the mouse-cursor position.

However, because some of the functions are not made virtual, it does not draw anything on the window.

Do not modify the main function. Make some functions virtual so that the program works as described above.

Also use const qualifiers in appropriate places.

(You only need very small modifications.)

```
#include <stdio.h>
#include "fssimplewindow.h"

class Drawable
{
public:
    void Draw(void);
};

class Movable : public Drawable
{
public:
    int x,y;
    void Move(void);
    void SetPosition(int xx,int yy);
};

class MouseChaser : public Movable
{
public:
    void Move(void);
    void Draw(void);
};

class MovableByArrowKeys : public Movable
{
public:
    void Move(void);
    void Draw(void);
};

void Drawable::Draw(void)
{
}

void Movable::Move(void)
{
}

void Movable::SetPosition(int xx,int yy)
{
    x=xx;
    y=yy;
}

void MouseChaser::Move(void)
{
    int lb,mb,rb,mx,my;
    FsGetMouseEvent(lb,mb,rb,mx,my);
```

```
    if(x<mx-3)
    {
        x+=3;
    }
    else if(mx+3<x)
    {
        x-=3;
    }
    if(y<my-3)
    {
        y+=3;
    }
    else if(my+3<y)
    {
        y-=3;
    }
}

void MouseChaser::Draw(void)
{
    glColor3ub(0,255,0);
    glBegin(GL_LINE_LOOP);
    glVertex2i(x-10,y);
    glVertex2i(x,y-10);
    glVertex2i(x+10,y);
    glVertex2i(x,y+10);
    glEnd();
}

void MovableByArrowKeys::Move(void)
{
    if(0!=FsGetKeyState(FSKEY_LEFT))
    {
        x-=5;
    }
    if(0!=FsGetKeyState(FSKEY_RIGHT))
    {
        x+=5;
    }
    if(0!=FsGetKeyState(FSKEY_UP))
    {
        y-=5;
    }
    if(0!=FsGetKeyState(FSKEY_DOWN))
    {
        y+=5;
    }
}

void MovableByArrowKeys::Draw(void)
{
    glColor3ub(0,0,255);
    glBegin(GL_QUADS);
    glVertex2i(x-10,y-10);
    glVertex2i(x+10,y-10);
    glVertex2i(x+10,y+10);
    glVertex2i(x-10,y+10);
    glEnd();
}

int main(void)
{
    FsOpenWindow(16,16,800,600,1);

    MouseChaser diamond;
    MovableByArrowKeys square;

    const int nMovable=2;
    Movable *movables[nMovable]={&diamond,&square};

    for(int i=0; i<nMovable; ++i)
    {
```

```
        movables[i]->SetPosition(400,300);
    }

    for(;;)
    {
        FsPollDevice();

        const int key=FsInkey();
        if(FSKEY_ESC==key)
        {
            break;
        }

        for(int i=0; i<nMovable; ++i)
        {
            movables[i]->Move();
        }

        glClear(GL_DEPTH_BUFFER_BIT|GL_COLOR_BUFFER_BIT);
        for(int i=0; i<nMovable; ++i)
        {
            movables[i]->Draw();
        }
        FsSwapBuffers();

        FsSleep(25);
    }

    return 0;
}
```