# Code:

BarGraph.h:

class BarGraph

{

    int size;

    char symbol;

public:

    BarGraph() {}

    ~BarGraph() {}

    void setSize(int sz);

    void setSymbol(char sym);

    static void displayAxis(char sym);

    void getBar();

};

BarGraph.cpp:

#include "BarGraph.h"

#include <iostream>

using std::cout;

using std::endl;

void BarGraph::setSize(int sz)

{

    size = sz;

}

void BarGraph::setSymbol(char sym)

{

    symbol = sym;

}

void BarGraph::displayAxis(char sym)

{

    cout << "2  10   20   30   40   50   60   70   80   90   100%" << endl;

    cout << "|  |    |    |    |    |    |    |    |    |    |   " << endl;

    for (int i = 0; i < 49; i++)

    {

        cout << sym;

    }

    cout << endl;

}

void BarGraph::getBar()

{

    for (int i = 0; i < size; i++)

    {

        cout << symbol;

    }

    cout << endl;

}

GradeGraph.h:

class GradeGraph

{

    int nAs;

    int nBs;

    int nCs;

    int nDs;

    int nFs;

public:

    GradeGraph() {}

    ~GradeGraph() {}

    void set(int numAs, int numBs, int numCs, int numDs, int numFs);

    void draw();

};

GradeGraph.cpp:

#include "GradeGraph.h"

#include "BarGraph.h"

#include <math.h>

void GradeGraph::set(int numAs, int numBs, int numCs, int numDs, int numFs)

{

    nAs = numAs;

    nBs = numBs;

    nCs = numCs;

    nDs = numDs;

    nFs = numFs;

}

void GradeGraph::draw()

{

    BarGraph::displayAxis('+');

    BarGraph \*BarA = new BarGraph;

    BarGraph \*BarB = new BarGraph;

    BarGraph \*BarC = new BarGraph;

    BarGraph \*BarD = new BarGraph;

    BarGraph \*BarF = new BarGraph;

    int sum = nAs + nBs + nCs + nDs + nFs;

    BarA->setSize(ceil(nAs / (sum / 50.0)));

    BarA->setSymbol('A');

    BarA->getBar();

    BarB->setSize(ceil(nBs / (sum / 50.0)));

    BarB->setSymbol('B');

    BarB->getBar();

    BarC->setSize(ceil(nCs / (sum / 50.0)));

    BarC->setSymbol('C');

    BarC->getBar();

    BarD->setSize(ceil(nDs / (sum / 50.0)));

    BarD->setSymbol('D');

    BarD->getBar();

    BarF->setSize(ceil(nFs / (sum / 50.0)));

    BarF->setSymbol('F');

    BarF->getBar();

}

PassFailGraph.h:

class PassFailGraph

{

    int nPass;

    int nFail;

public:

    PassFailGraph() {}

    ~PassFailGraph() {}

    void setNPass(int n);

    void setNFail(int n);

    int getNPass();

    int getNFail();

    void draw();

};

PassFailGraph.cpp:

#include "PassFailGraph.h"

#include "BarGraph.h"

#include <math.h>

void PassFailGraph::setNPass(int n)

{

    nPass = n;

}

void PassFailGraph::setNFail(int n)

{

    nFail = n;

}

int PassFailGraph::getNPass()

{

    return nPass;

}

int PassFailGraph::getNFail()

{

    return nFail;

}

void PassFailGraph::draw()

{

    BarGraph::displayAxis('=');

    BarGraph \*BarPass = new BarGraph;

    BarGraph \*BarFail = new BarGraph;

    int sum = nPass + nFail;

    BarPass->setSize(ceil(getNPass() / (sum / 50.0)));

    BarPass->setSymbol('P');

    BarPass->getBar();

    BarFail->setSize(ceil(getNFail() / (sum / 50.0)));

    BarFail->setSymbol('F');

    BarFail->getBar();

}

main.cpp:

#include <iostream>

#include "GradeGraph.h"

#include "PassFailGraph.h"

using std::cin;

using std::cout;

using std::endl;

// Include all the necessary libraries/files

int main()

{

    int nAs, nBs, nCs, nDs, nFs; // Variables representing input grades.

    cout << "Welcome to the grade grapher." << endl;

    cout << "Enter the number of As, Bs, Cs, Ds and Fs." << endl;

    cout << "As: ";

    cin >> nAs;

    cout << "Bs: ";

    cin >> nBs;

    cout << "Cs: ";

    cin >> nCs;

    cout << "Ds: ";

    cin >> nDs;

    cout << "Fs: ";

    cin >> nFs;

    // A object is created which sets and calls the PassFailGraph.

    PassFailGraph \*passFailGraph = new PassFailGraph;

    passFailGraph->setNPass(nAs + nBs + nCs);

    passFailGraph->setNFail(nDs + nFs);

    passFailGraph->draw();

    // A object is created which sets and calls the GradeGraph.

    GradeGraph \*gradegraph = new GradeGraph;

    gradegraph->set(nAs, nBs, nCs, nDs, nFs);

    gradegraph->draw();

    // delete objects

    delete passFailGraph;

    delete gradegraph;

}

Makefile:

OBJS = BarGraph.o GradeGraph.o PassFailGraph.o main.o

main: $(OBJS)

    g++ -o main $(OBJS)

%o: %.cpp %.h

    g++ -c $<

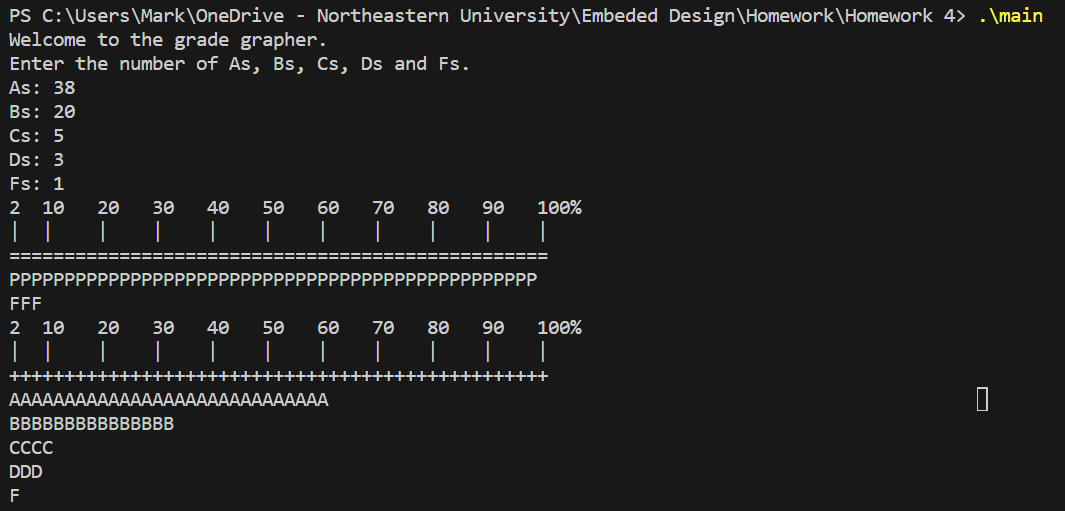
.PHONY: clean

clean:

    rm -f $(OBJS).o

# Sample Outputs:

Case 1:



Case 2:

