**GradedActivity.h**

#ifndef GRADEDACTIVITY\_H

#define GRADEDACTIVITY\_H

// GradedActivity class declaration

class GradedActivity

{

private:

double score; // To hold the numeric score

public:

// Default constructor

GradedActivity()

{ score = 0.0; }

// Constructor

GradedActivity(double s)

{ score = s; }

// Mutator function

void setScore(double s)

{ score = s; }

// Accessor functions

double getScore() const

{ return score; }

char getLetterGrade() const;

};

#endif

**GradedActivity.cpp**

#include "GradedActivity.h"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Member function GradedActivity::getLetterGrade \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

char GradedActivity::getLetterGrade() const

{

char letterGrade; // To hold the letter grade

if (score > 89)

letterGrade = 'A';

else if (score > 79)

letterGrade = 'B';

else if (score > 69)

letterGrade = 'C';

else if (score > 59)

letterGrade = 'D';

else

letterGrade = 'F';

return letterGrade;

}

**FinalExam.h**

#ifndef FINALEXAM\_H

#define FINALEXAM\_H

#include "GradedActivity.h"

class FinalExam : public GradedActivity

{

private:

int numQuestions; // Number of questions

double pointsEach; // Points for each question

int numMissed; // Number of questions missed

public:

// Default constructor

FinalExam()

{ numQuestions = 0;

pointsEach = 0.0;

numMissed = 0; }

// Constructor

FinalExam(int questions, int missed)

{ set(questions, missed); }

// Mutator function

void set(int, int); // Defined in FinalExam.cpp

// Accessor functions

double getNumQuestions() const

{ return numQuestions; }

double getPointsEach() const

{ return pointsEach; }

int getNumMissed() const

{ return numMissed; }

};

#endif

**FinalExam.cpp**

#include "FinalExam.h"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// set function \*

// The parameters are the number of questions and the \*

// number of questions missed. \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void FinalExam::set(int questions, int missed)

{

double numericScore; // To hold the numeric score

// Set the number of questions and number missed.

numQuestions = questions;

numMissed = missed;

// Calculate the points for each question.

pointsEach = 100.0 / numQuestions;

// Calculate the numeric score for this exam.

numericScore = 100.0 - (missed \* pointsEach);

// Call the inherited setScore function to set

// the numeric score.

setScore(numericScore);

}

**ExamDriver.cpp**

// This program demonstrates a base class and a derived class.

#include <iostream>

#include <iomanip>

#include "FinalExam.h"

using namespace std;

int main()

{

int questions; // Number of questions on the exam

int missed; // Number of questions missed by the student

// Get the number of questions on the final exam.

cout << "How many questions are on the final exam? ";

cin >> questions;

// Get the number of questions the student missed.

cout << "How many questions did the student miss? ";

cin >> missed;

// Define a FinalExam object and initialize it with

// the values entered.

FinalExam test(questions, missed);

// Display the test results.

cout << setprecision(2);

cout << "\nEach question counts " << test.getPointsEach()

<< " points.\n";

cout << "The exam score is " << test.getScore() << endl;

cout << "The exam grade is " << test.getLetterGrade() << endl;

return 0;

}