

```

//*****
// Division Program
// Divisend and divisor are prompted for and read.
// If divisor is 0, division is not performed;
// otherwise, division is performed and result is printed.
//*****

```

```

#include <iostream>
using namespace std;

```

```

int main()
{
    int dividend;
    int divisor;
    int result;
    cout << "Enter dividend and divisor" << endl;
    cin >> dividend >> divisor;

    if (divisor != 0)
    {
        result = dividend / divisor;
        cout << "Result is " << result << endl;
    }
    else
    {
        cout << "Division by zero is not allowed." << endl;
        result = 9999;
    }

    system("PAUSE");
    return 0;
}

```

```

-----
// This program uses an if/else if statement to assign a
// letter grade (A, B, C, D, or F) to a numeric test score.
// It validates the user's input using try - catch exception handling routine
// and also validates scores within a specified range
#include <iostream>
#include <cstdlib>
#include <stdexcept>    // for exception, runtime_error, out_of_range
using namespace std;

```

```

int main()
{
    int testScore;    // Holds a numeric test score
    char grade;       // Holds a letter grade

    // Get the numeric score
    cout << "Enter your numeric test score and I will\n";
    cout << "tell you the letter grade you earned: ";
    cin >> testScore;

    /*
    if(!cin){
        cerr <<"INCORRECT INPUT"<<endl;
        //keep_window_open();
        return 1;
    }*/

    // a simple exception handling routine
    try{
        if(!cin)
            throw runtime_error ("an input error");
    }
    catch (runtime_error &e) {
        cout << "Caught a runtime_error exception: "
             << e.what () << '\n';
        system("PAUSE");
        exit(1);
    }

    if ((testScore < 0) || (testScore > 100)) // Input validation
    { // testScore is invalid
        cout << testScore << " is an invalid score.\n";
        cout << "Run the program again and enter a value\n";
        cout << "in the range of 0 to 100.\n";
    }
    else
    { // testScore is valid so determine the letter grade
        if (testScore < 60)
            grade = 'F';
        else if (testScore < 70)
            grade = 'D';
        else if (testScore < 80)
            grade = 'C';
        else if (testScore < 90)
            grade = 'B';
        else // If we got this far, testScore must be >= 90

```

```

        grade = 'A';

        // Display the letter grade
        cout << "Your grade is " << grade << endl;
    }

    return 0;
}

```

```

//*****
// Notices program: This program determines (1) a student's average based on three
// test scores and (2) the student's passing/failing status
//*****
#include <iostream>
#include <iomanip>    // For setprecision()

```

```
using namespace std;
```

```

int main()
{
    float average;        // Average of three test scores
    long studentID;       // Student's identification number
    int test1;            // Score for first test
    int test2;            // Score for second test
    int test3;            // Score for third test
    bool dataOK;          // True if data is correct

    cout << fixed << showpoint;        // Set up floating pt.
                                         // output format

    // Get data

    cout << "Enter a Student ID number and three test scores:"
         << endl;
    cin >> studentID >> test1 >> test2 >> test3;
    cout << "Student number: " << studentID << " Test Scores: "
         << test1 << ", " << test2 << ", " << test3 << endl;

    // Test data

    if (test1 < 0 || test2 < 0 || test3 < 0)
        dataOK = false;
    else
        dataOK = true;

    if (dataOK)
    {
        // Calculate average

        average = float(test1 + test2 + test3) / 3.0;

        // Print message

        cout << "Average score is "
             << setprecision(2) << average << "--";
        if (average >= 60.0)
        {
            cout << "Passing";                // Student is passing
            if (average < 70.0)
                cout << " but marginal";      // But marginal
            cout << "." << endl;
        }
        else
            cout << "Failing." << endl;        // Student is failing
    }
    else
        // Invalid data
        cout << "Invalid Data: Score(s) less than zero." << endl;
    system ("PAUSE");
    return 0;
}

```

```

-----
// Program Area demonstrates stream testing
#include <iostream>
#include <fstream>
using namespace std;

```

```

int main()
{
    int side1;            // one side of a rectangle
    int side2;            // the other side of a rectangle
    ifstream inData;      // file stream
    int area;             // area of rectangle

    inData.open("myData.dat");
    if (!inData)
    {

```

```

        cout << "Input file not found." << endl;
        return 1;
    }
    inData >> side1 >> side2;
    if (!inData)
    {
        cout << "Data format incorrect.";
        return 2;
    }
    area = side1 * side2;
    cout << "Area is " << area << endl;
    return 0;
}

```

CONTENTS OF DATA FILE myData.dat  
5 5

---

```

/*Program LowScore reads data from an input file and prints three test scores.
The lowest value of the three is printed with an appropriate message.
Assumption: You are to create a data file with three scores and that the three scores are unique.*/
#include <iostream>

```

```

using namespace std;

```

```

int main ()
{
    int test1Score;
    int test2Score;
    int test3Score;
    //add fstream variables

```

```

/* cout << "Enter score for test 1; press return." << endl;
cin >> test1Score;
cout << "Enter score for test 2; press return." << endl;
cin >> test2Score;
cout << "Enter score for test 3; press return." << endl;
cin >> test3Score;*/

```

```

/*WRITE CODE TO OPEN AND READ DATA FROM FILE*/
/*VALIDATE THE INPUT FILE STREAM TO CHECK IF DATA FILE HAS BEEN
OPENED AND THAT THE DATA MATCHES THE VARIABLES IN WHICH THEY WILL BE STORED*/

```

```

//see CCCConfer Illuminiat "live" lab recording of 9/19/2012

```

```

cout << "The three test scores are: " << endl;
cout << test1Score << endl;
cout << test2Score << endl;
cout << test3Score << endl;

```

```

/*WRITE LOGICAL EXPRESSIONS IE., IF-THEN-ELSE STATEMENTS TO DETERMINE LOWEST SCORE*/

```

```

system("PAUSE");

```

```

return 0;

```

```

}

```