5a

#include <iostream>

using namespace std;

// Function prototypes

double getLength();

double getWidth();

double getArea(double, double);

void displayData(double, double, double);

int main()

{

double length; // To hold the rectangle's length

double width; // To hold the rectangle's width

double area; // To hold the rectangle's area

// Get the rectangle's length.

length = getLength();

// Get the rectangle's width.

width = getWidth();

// Get the rectangle's area.

area = getArea(length, width);

// Display the rectangle's data.

displayData(length, width, area);

return 0;

}

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

// getLength function \*

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

double getLength()

{

double length; // To hold the length

// Get the length.

cout << "Enter the length: ";

cin >> length;

// Return the length.

return length;

}

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

// getWidth function \*

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

double getWidth()

{

double width; // To hold the width

// Get the width.

cout << "Enter the width: ";

cin >> width;

// Return the width.

return width;

}

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

// getArea function \*

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

double getArea(double length, double width)

{

// Return the area.

return length \* width;

}

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

// displayData function \*

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

void displayData(double length, double width, double area)

{

cout << "\nRectangle Data\n"

<< "--------------\n"

<< "Length: " << length << endl

<< "Width: " << width << endl

<< "Area: " << area << endl;

}

Text

Description automatically generated

///////////////////////////////////////////////////////////////////////////////////////////////////

5b

#include <iostream>

#include <cmath>

using namespace std;

const double g=9.8;

double falling\_distance(double);

int main()

{

double distance,time;

for (int a= 0; a < 10;a++)

{

time = a + 1;

distance = falling\_distance(time);

cout << time<< " seconds= "<<distance<<"meters"<<endl;

}

return 0;

}

double falling\_distance(double falling\_time)

{

return (0.5)\*g\*pow(falling\_time, 2);

}

Text

Description automatically generated

/////////////////////////////////////////

5c

#include <iostream>

using namespace std;

double input(double);

double Lowest(double, double, double, double, double);

double lowest(double, double, double, double, double);

void score(double &);

void average(double, double, double, double, double);

int main()

{

double score1,score2,score3,score4,score5;

score(score1);

score(score2);

score(score3);

score(score4);

score(score5);

average(score1,score2,score3,score4,score5);

return 0;

}

double input(double num)

{

while(!(cin >> num) || (num < 0 || num > 100))

{

cout << "Error. An integer from 0-100 must be entered: ";

cin.clear();

cin.ignore();

}

return num;

}

void score(double &num)

{

cout << "Please enter testscores ";

num = input(num);

}

void average(double num1,double num2,double num3,double num4,double num5)

{

double average1,

lowest1 = lowest(num1, num2, num3, num4, num5);

if (num1 == lowest1)

average1=(num1+num2+num3+num4)/4;

else if (num2 == lowest1)

average1=(num1+num2+num3+num4)/4;

else if (num3 == lowest1)

average1=(num1+num2+num3+num4)/4;

else if (num4 == lowest1)

average1=(num1+num2+num3+num4)/4;

else if (num5 == lowest1)

average1=(num1+num2+num3+num4)/4;

cout<<"The average is: "<<average<<endl;

}

double Lowest(double num1,double num2,double num3,double num4,double num5)

{

double lowest1;

if (num1 <= num2)

{

if (num1 <= num3)

{

if (num1 <= num4)

{

if (num1 <= num5)

{

lowest1 = num1;

}

}

}

}

return lowest1;

}

double lowest(double num1,double num2,double num3,double num4,double num5)

{

double smallest;

smallest = Lowest(num1, num2, num3, num4, num5);

smallest = Lowest(num2, num3, num4, num5, num1);

smallest = Lowest(num3, num4, num5, num1, num2);

smallest = Lowest(num4, num5, num1, num2, num3);

smallest = Lowest(num5, num1, num2, num3, num4);

return smallest;

}

Text

Description automatically generated