Project 2

Bridgette Stranko

Jacob Tubman

Pseudo Code

Include IO Stream

Include CMath

Using namespace std;

Comment: UDF to get the four scores and calculates average and standard deviation

Comment: Returns all 6 parameters

void calculate\_scores(return int S1, return int S2, return int S3, return int S4, return double Avg, return double Std\_Deviation);

Comment: UDF to get user input for the four scores

Comment: Returns all 4 parameters

void get\_input(return int Score1, return int Score2, return int Score3, return int Score4);

Comment: UDF to format the results for highest and lowest score, the average, and the standard deviation

Comment: Display only. Nothing is returned

void show\_results(int Min, int Max, double Avg, double Std\_Dev);

Comment: UDF to find minimum and maximum using four scores from the function that called it

Comment: Returns only the minimum and maximum

void get\_min\_max(int S1, int S2, int S3, int S4, return int Min, return int Max);

Comment: UDF to order the scores from the function that called it

Comment: Display only. Nothing is returned

void order\_scores(int S1, int S2, int S3, int S4);

Comment: UDF to swap values

Comment: The new values are returned

void swap\_values(return int value1, return int value2);

int main ()

{

Comment: Declare variables

char ans; Comment: used for do while loop

int score1, score2, score3, score4, min, max;

double average, std\_deviation;

do

{

Comment: Sets the precision to hundreths

To Screen: .setf(ios::fixed);

To Screen: .setf(ios::showpoint);

To Screen: .precision(2);

Comment: Retrieves user input and calculates the average and standard deviation

calculate\_scores(score1, score2, score3, score4, average, std\_deviation);

Comment: Find the lowest and highest numbers

get\_min\_max(score1, score2, score3, score4, min, max);

Comment: Display the scores in the order they were entered

To Screen: << "(new line)For the scores " << score1 << ", " << score2 << ", " << score3

<< ", and " << score4 << ": "<<end line << end line;

Comment: Display the scores in order

order\_scores(score1, score2, score3, score4);

Comment: Displays the lowest score, highest score, the average and standard deviation

show\_results(min, max, average, std\_deviation);

To Screen: << "Would you like to run the program again? (new line)"

<< " (Type Y for yes): ";

Comment: User lets the program know if they would like to run again

From Screen >> ans;

Comment: Formats a new line between runs

To Screen: << end line;

} while (ans == 'Y' or ans == 'y');

Comment: end do while loop if ans does not equal Y or y

Comment: Program will terminate if any other character or integer is entered

Comment: return 0 to main

return 0;

}

Comment: Function definitions for the UDF declared above main

void calculate\_scores(return int S1, return int S2, return int S3, return int S4, return double Avg, return double Std\_Deviation)

{

Comment: call UDF get input and return 4 scores

get\_input (S1, S2, S3, S4);

Comment: calculates the average

Avg = (S1 + S2 + S3 + S4)/4.0;

Comment: calculates the standard deviation

Std\_Deviation = sqrt((pow((S1-Avg), 2)+ pow((S2-Avg), 2)+ pow((S3-Avg), 2) +

pow((S4-Avg), 2))/4);

}

void get\_input(return int Score1, return int Score2, return int Score3, return int Score4)

{

Comment: request input from user

To Screen: <<"(new line)Enter four scores. Please round to the nearest whole number.(new line)"

<<"Don't forget to put a space between the scores: ";

Comment: user input for the four scores

From Screen >> Score1 >> Score2 >> Score3 >> Score4;

Comment: all four scores are returned

return;

}

void show\_results(int Min, int Max, double Avg, double Std\_Dev)

{

Comment: displays the lowest score, highest score, average and standard deviation

To Screen: << "Your lowest score was " << Min << " and your highest score was " <<Max << end line

<< "The average is " << Avg << " and the standard deviation is " << Std\_Dev

<<end line <<end line;

Comment: Display only nothing is returned

}

void get\_min\_max(int S1, int S2, int S3, int S4, return int Min, return int Max)

{

Comment: If S1 is the lowest number then Min is assigned the value of S1

if(S1 <= S2 and S1 <= S3 and S1 <= S4)

Min = S1;

Comment: If S2 is the lowest number then Min is assigned the value of S2

else if (S2 <= S1 and S2 <= S3 and S2 <= S4)

Min = S2;

Comment: If S3 is the lowest number then Min is assigned the value of S3

else if (S3 <= S1 and S3 <= S2 and S3 <= S4)

Min = S3;

Comment: If all above Booleans are false then S4 must be the lowest number

Comment: Min is assigned the value of S4

else

Min = S4;

Comment: If S1 is the highest number then Max is assigned the value of S1

if (S1 >= S2 and S1 >= S3 and S1 >= S4)

Max = S1;

Comment: If S2 is the highest number then Max is assigned the value of S2

else if (S2 >= S1 and S2 >= S3 and S2 >= S4)

Max = S2;

Comment: If S3 is the highest number then Max is assigned the value of S3

else if (S3 >= S1 and S3 >= S2 and S3 >= S4)

Max = S3;

Comment: If all above Booleans are false then S4 must be the highest number

Comment: Max is assigned the value of S4

else

Max = S4;

}

void order\_scores(int S1, int S2, int S3, int S4)

{

Comment: Check to see if S1 is greater than S2, S3 and then S4

Comment: If S1 is greater than any of these then swap the values

if(S1 > S2)

swap\_values(S1, S2);

if(S1 > S3)

swap\_values(S1, S3);

if(S1 > S4)

swap\_values(S1, S4);

Comment: Check to see if S2 is greater than S3 and then S4

Comment: If S1 is greater than any of these then swap the values

Comment: Do not need to check S1. S1 is already the lowest number

if(S2 > S3)

swap\_values(S2, S3);

if(S2 > S4)

swap\_values(S2, S4);

Comment: Check to see if S3 is greater than S4

Comment: If S3 is greater than S4 then swap the values

Comment: Do not need to check S1 and S2 they are already in order

if(S3 > S4)

swap\_values(S3, S4);

Comment: The values have been switched and the scores are now in order

Comment: shows the scores in order

To Screen: << "The Scores in order are " << S1

<< ", " << S2 << ", " << S3 << ", " << S4 << end line;

Comment: Display only. Nothing is returned

return;

}

void swap\_values(return int value1, return int value2)

{

Comment: Declare a temporary place holder

int temp;

Comment: swap the values

temp = value1; Comment: temp = value1

value1 = value2; Comment: value1 now equals value2

value2 = temp; Comment: value2 now equals temp (which was holding value1)

}