**PROGRAM OUTPUT FORMATTING AND A PREVIEW OF FUNCTIONS DESIGNED TO WORK WITH STRING DATA TYPES**

**C++ MANIPULATORS for formatting output: setw(n), setprecision(n), fixed, showpoint, left and right.**

**  
C++ functions for processing string data: s.length( ), s.find(str, x ) and s.substr(x, n)**

****

Files used to demonstrate topics using a C++ compiler

setwidth1.cpp & setwidth2.cpp - demonstrates output format using setw(n) manipulator made available with the #include<iomanip> header file

setprecision1.cpp, setprecision2.cpp - demonstrates output format using setprecision(n) manipulator made available with the #include<iomanip> header file

setprecision\_fixed.cpp - demonstrates what happens when fixed and setprecision (n) manipulators are used together.

setprecision\_showpoint.cpp - demonstrates what happens when showpoint and setprecision (n) manipulators are used together.

setwidth\_left\_right\_manipulators.cpp - demonstrates how to use manipulators to right and left justify output.

format.cpp - demonstrates the use of field width specifications.

Strdemo.cpp - demonstrates string functions.

//This program (setwidth1.cpp) displays two rows of numbers and demonstrates the use of the setw (fieldwidth) manipulator  
#include <iostream> //includes fixed, showpoint and left manipulators  
#include <iomanip> //required for setprecision and setw

using namespace std;

int main()

{

int value;

int value1 =1234, value2 = 6, value3 =900, value4 =12,value5=5, value6= 2345;

//display the first two rows of values

cout<<value1<<value2<<value3<<endl;

cout<<value4<<value5<<value6<<endl;

//display two rows use the setw(fieldwidth) manipulator

cout<<setw(6)<<value1<<setw(6)<<value2<<setw(6)<<value3<<endl;

cout<<setw(6)<<value4<<setw(6)<<value5<<setw(6)<<value6<<endl;

//another setw use

value=23134;

cout<<setw(2)<<value<<endl;

return 0;

}

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

// This program (setwidth2.cpp) demonstrates the setw manipulator used with values of various data types.

#include <iostream> //includes fixed, showpoint and left manipulators

#include <iomanip> //required for setprecision and setw

using namespace std;

int main()

{

int intValue = 3456;

double doubleValue = 92.5;

string fname = "Seth Evans Sarkar";

cout << setw(5) << intValue << endl;

cout << setw(8) << doubleValue << endl;

cout << setw(16) << fname << endl;

return 0;

}

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

// This program (setprecision1.cpp) demonstrates how setprecision rounds a floating point value.

#include <iostream>

#include <iomanip> //required for setprecision

using namespace std;

int main()

{

double quotient, number1 = 260.654, number2 = 23.45;

quotient = number1 / number2;

cout << quotient << endl;

cout << setprecision(5) << quotient << endl;

cout << setprecision(4) << quotient << endl;

cout << setprecision(3) << quotient << endl;

cout << setprecision(2) << quotient << endl;

cout << setprecision(1) << quotient << endl;

return 0;

}

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

// This program (setprecision2.cpp) demonstrates how setprecision rounds a floating point value.

#include <iostream>

#include <iomanip> //Required for setprecision

using namespace std;

int main()

{

//section below shows how setprecision affects the way various values are displayed

double dollars= 42.56;

cout<<dollars<<endl; //displays 42.56

cout<<setprecision(5)<<dollars<<endl; //displays 42.56

double value1= 38.92789;

cout << setprecision(3) << value1 << endl; //value displayed 38.9

double value2= 63.34567;

cout << setprecision(2) << value2 << endl; //value displayed 63

int number=21;

cout << setprecision(5) << number << endl; //value displayed 21

float item=109.5;

cout << setprecision(4) << item << endl; //value displayed 109.5

return 0;

}

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

// This program (setprecision\_fixed.cpp) demonstrates what happens when fixed and setprecision manipulators are used together.

// The value specified by the setprecision manipulator will be the number of digits to appear after the decimal point,

// not the number of significant digits.

#include <iostream> //includes fixed, showpoint and left manipulators

#include <iomanip> //required for setprecision and setw

using namespace std;

int main()

{

cout<<fixed; //another stream manipulator forces cout to print the digits in fixed-point notation, or decimal

//section below shows how setprecision with the fixed manipulator affects the way various values are displayed

double x= 345.6, y=789.0;

cout<<setprecision(6)<<x<<endl; //displays 345.6

cout<<y<<endl; //displays 789

double value1= 38.92789;

cout << setprecision(3) << value1 << endl; //value displayed 38.928

double value2= 63.34567;

cout << setprecision(2) << value2 << endl; //value displayed 63.35

int number=21;

cout << setprecision(5) << number << endl; //value displayed 21

float item=109.5;

cout << setprecision(4) << item << endl; //value displayed 109.5000

return 0;

}

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

// This program (setprecision\_showpoint.cpp) demonstrates what happens when showpoint and setprecision manipulators are used together.

#include <iostream> //includes fixed, showpoint and left manipulators

#include <iomanip> //required for setprecision and setw

using namespace std;

int main()

{

//section below shows how setprecision without and with showpoint manipulator affects the way various values are displayed

double x= 345.6, y=789.0;

cout<<setprecision(6)<<x<<endl; //displays 345.6..note no trailing zeroes indicates that 6 significant digits were ignored

cout<<y<<endl; //displays 789..note no trailing zeroes indicates that 6 significant digits were ignored

cout<<showpoint; //diplays decimal point and trailing zeroes for floating point numbers

cout<<setprecision(6)<<x<<endl; //displays 345.600..notice now we have 6 significant digits display

cout<<y<<endl; //displays 789.000..notice now we have 6 significant digits display

system ("PAUSE");

return 0;

}

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

// This program (setwidth\_left\_right\_manipulators.cpp) demonstrates how to use manipulators to right and left justify output.

#include <iostream> //includes fixed, showpoint and left manipulators

#include <iomanip> //required for setprecision and setw

using namespace std;

int main()

{

//section below shows how to use manipulators to right and left justify spaces when using setw

double x= 346.678, y=34.3, z=1.678;

//use the setw manipulator to display variables x,y,z right-justified (default)in a print field of 10 spaces on seperate lines

cout<<setw(10)<<x<<endl;

cout<<setw(10)<<y<<endl;

cout<<setw(10)<<z<<endl;

cout<<left; //set the left manipulator to left justify spaces when using setw

//use the setw manipulator to display variables x,y,z left-justified in a print field of 10 spaces on seperate lines

cout<<setw(10)<<x<<endl;

cout<<setw(10)<<y<<endl;

cout<<setw(10)<<z<<endl;

return 0;

}

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

// Program format.cpp demonstrates the use of fieldwidth specifications.

#include <iostream>

#include <iomanip>

using namespace std;

const int INT\_NUMBER = 1066;

const float FLT\_NUMBER = 3.14159;

int main ()

{

//variables declared

float fltValue;

int intValue;

double dblValue;

//variables assignments with type casting within arithmetic expressions

intValue = INT\_NUMBER + int(FLT\_NUMBER); //value assigned to variable is 1069

fltValue = float(INT\_NUMBER) + FLT\_NUMBER; //value assigned to variable is 1069.14159

dblValue = double (INT\_NUMBER) + double(FLT\_NUMBER); //value assigned to variable is 1069.14159

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

//diplay showing various output formats

cout << INT\_NUMBER << endl;

cout << intValue << endl;

cout << setw(10) << intValue;

cout << setw(10) << intValue << intValue /10 << endl;

cout << fltValue << endl; //interesting precision output

cout << setprecision(10) << fltValue << endl;

cout << dblValue << endl; //interesting precision output

cout << setw(10) << setprecision(3) << fltValue << endl;

cout << fltValue << endl;

cout << intValue << setw(3) << intValue << setw(7) << intValue << endl;

fltValue = 123456789.6;

cout << fltValue << endl; //another interesting output

cout.unsetf(ios::fixed | ios::showpoint); // turns off fixed and showpoint

cout << fltValue << endl;

system("PAUSE");

return 0;

}

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

// Program strdemo.cpp demonstrates string functions.

#include <iostream>

#include <string>

using namespace std;

const string TITLE = "How much was the doggie in the window?";

const string CAT = "cat";

int main()

{

cout << TITLE.length() << endl;

cout << TITLE.find("the") << endl;

cout << TITLE.find(CAT) << endl;

cout << TITLE.substr(17, 6) << endl;

cout << TITLE.substr(17, 23) << endl;

cout << TITLE.substr(17, 23).length() << endl;

return 0;

}