CSCI 152 Programming Fundamentals II Spring 2018

Dr. Creider Program to illustrate type concepts-2

/\* CSCI 152 Spring 2018

program to demonstrate some type concepts, over flow and under flow

Run the program several times with different input values

Examine the results produced by the statements to determine what happened

\*/

#include<iostream>

using namespace std;

int main()

{

short num1S, num2S, resultS, stop=1; // declare a signed variable of 2 bytes: range -32768 to 32767

unsigned short num1US, num2US, resultUS; // declare an unsigned variable of 2 bytes: range 0 to 65535

long num1L, num2L, resultL; // declare a signed number of 4 bytes: range of -2147483648 to 2147483647

long long resultLL1, resultLL2; // declare a signed number of 4 bytes: range -9223372036854775808 to

9223372036854775807

while(stop)

{

// example with signed short

cout<<"enter two short numbers in the range of -32768 to 32767: ";

cin>>num1S>>num2S;

resultS = num1S \* num2S;

num1L = num1S \* num2S;

num2L = num1S \* static\_cast<long>(num2S);

cout<<"muliplying two short variables - resultS = "<<resultS<<"\n\tnum1L = "<<num1L<<", num2L = "<<num2L<<endl;

resultS = num1S + num2S;

num1L = num1S + num2S;

num2L = num1S + static\_cast<long>(num2S);

cout<<"adding two short variables - resultS = "<<resultS<<"\n\tnum1L = "<<num1L<<", num2L = "<<num2L<<endl;

resultS = num1S - num2S;

num1L = num1S - num2S;

num2L = num1S - static\_cast<long>(num2S);

cout<<"subtracting two short variables - resultS = "<<resultS<<"\n\tnum1L = "<<num1L<<", num2L = "<<num2L<<endl;

// example with unseigned short

cout<<"\nenter two short numbers in the range of 0 to 65535: ";

cin>>num1US>>num2US;

resultUS = num1US \* num2US;

num1L = num1US \* num2US;

num2L = num1US \* static\_cast<long>(num2US);

cout<<"muliplying two short variables - resultUS = "<<resultUS<<"\n\tnum1L = "<<num1L<<", num2L = "<<num2L<<endl;

resultUS = num1US + num2US;

num1L = num1US + num2US;

num2L = num1US + static\_cast<long>(num2US);

cout<<"adding two short variables - resultUS = "<<resultUS<<"\n\tnum1L = "<<num1L<<", num2L = "<<num2L<<endl;

resultUS = num1US - num2US;

num1L = num1US - num2US;

num2L = num1US - static\_cast<long>(num2US);

cout<<"subtracting two short variables - resultuS = "<<resultUS<<"\n\tnum1L = "<<num1L<<", num2L = "<<num2L<<endl;

// example mixing signed and unsigned

resultS = num1US;

resultUS = num1S;

cout<<"\nstoring an unsigned short into a signed short - nums1US = "<<num1US<<", resultS = "<<resultS<<endl;

cout<<"storing an signed short into a unsigned short - nums1S = "<<num1S<<", resultUS = "<<resultUS<<endl;

// example using signed long

cout<<"\nenter two long numbers in the range of -2147483648 to 2147483647:\n";

cin>>num1L>>num2L;

resultL = num1L \* num2L;

resultLL1 = num1L \* num2L;

resultLL2 = num1L \* static\_cast<long long>(num2L);

cout<<"muliplying two long variables - resultL = "<<resultL<<"\n\tresultLL1 = "<<resultLL1<<", resultLL2 = "<<resultLL2<<endl;

resultL = num1L + num2L;

resultLL1 = num1L + num2L;

resultLL2 = num1L + static\_cast<long long>(num2L);

cout<<"adding two long variables - resultL = "<<resultL<<"\n\tresultLL1 = "<<resultLL1<<", resultLL2 = "<<resultLL2<<endl;

resultL = num1L - num2L;

resultLL1 = num1L - num2L;

resultLL2 = num1L - static\_cast<long long>(num2L);

cout<<"subtracting two long variables - resultL = "<<resultL<<"\n\tresultLL1 = "<<resultLL1<<", resultLL2 = "<<resultLL2<<endl;

cout<<"\nTo stop this program enter 0 else enter any other short number ";

cin>>stop;

}

// pause the program to see the results

system("pause"); // this is a command to the Windows operating system

//return 0; // statement not necessary in Bloodshed compiler

}