**Computer Engineering**

**CE102**

**Lab Homework 6**

**Questions:**

1. For each of the following, write a single statement that performs the specified task. Assume that long variables value1 and value2 have been declared and value1 has been initialized to 200000.

a) Declare the variable longPtr to be a pointer to an object of type long.

b) Assign the address of variable value1 to pointer variable longPtr.

c) Display the value of the object pointed to by longPtr.

d) Assign the value of the object pointed to by longPtr to variable value2.

e) Display the value of value2.

f) Display the address of value1.

g) Display the address stored in longPtr. Is the address displayed the same as value1’s?

#include <iostream>

using namespace std;

int main()

{

long value1 = 200000;

long value2;

//a:

long \*longPtr;

//b:

longPtr = &value1;

//c:

cout << \*longPtr <<endl;

//d:

value2 = \*longPtr;

//e:

cout << value2 << endl;

//f:

cout << &value1 << endl;

//g: Yes, the address displayed the same value as value1's

cout << longPtr << endl;

}

1. Perform the task in each of the following statements:

a) Write the function header for function zero that takes a long integer built-in array parameter bigIntegers and does not return a value.

b) Write the function prototype for the function in part (a).

c) Write the function header for function add1AndSum that takes an integer built-in array parameter oneTooSmall and returns an integer.

d) Write the function prototype for the function described in part (c).

//a:

void zero(long int \*bigIntegers);

//b:

void zero(long int \*);

//c:

int add1AndSum(int \*oneTooSmall);

//d:

int add1AndSum(int \*);

1. Find the error in each of the following segments. If the error can be corrected, explain how.

***a) int \*number;***

***cout << number << endl;***

we can fix this by assigning the address of the variable x (I created this value) to pointer number

#include <iostream>

using namespace std;

int main()

{

int x;

int \*number = &x;

cout << number << endl;

}

***b) double \*realPtr;***

***long \*integerPtr;***

***integerPtr = realPtr;***

realPtr and integerPtr have different type of pointers. So, it gives an error when we write integerPtr = realPtr. Because double\* cannot convert to long int in assignment. We can fix it by using the same type of pointers for this assignment.

double \*realPtr;

double \*integerPtr;

integerPtr = realPtr;

or

long \*realPtr;

long \*integerPtr;

integerPtr = realPtr;

***c) int \* x, y;***

***x = y;***

variable y is not a pointer, so it gives an error. We can solve this problem like this:

int \*x,y;

x = &y;

or

int \*x, \*y;

x = y;

***d) char s[ ] = "this is a character array";***

***for ( ; \*s != '\0'; ++s)***

***cout << \*s << ' ';***

char s[ ] = "this is a character array";

char \*sPtr = s;

for ( ; \*sPtr != '\0'; ++sPtr)

cout << \*sPtr << ' ';

*or*

Using ++s is giving us an error.

char s [] = {"this is a character array"};

for (int t{0}; s[t] != '\0'; ++t)

cout << s[t] << ' ';

***e) short \*numPtr, result;***

***void \*genericPtr = numPtr;***

***result = \*genericPtr + 7;***

the compiler gives us an error like “error ‘\*void’ is not a pointer-to-object type “

short \*numPtr, result;

short \*genericPtr = numPtr;

result = \*genericPtr + 7;

***f) double x = 19.34;***

***double xPtr = &x;***

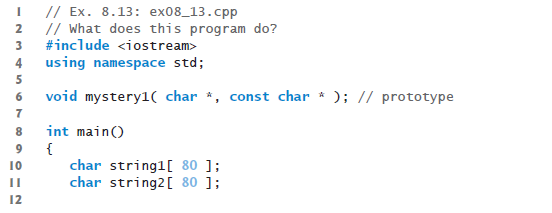
***cout << xPtr << endl;***

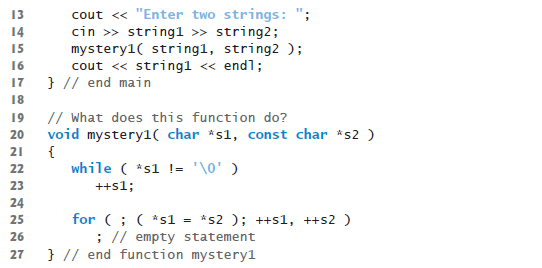
the compiler gives us an error like “error: cannot convert ‘double\*’ to ‘double’ in initialization”. According

to that we can solve this problem just fixing the second line of the code

double **\*xPtr** = &x;

1. What does the following program do?





Output:

Enter two strings: x

y

xy