{
    char * name;  // create pointer but no storage

    name = getname();  // assign address of string to name
    cout << name << " at " << (int *) name << "\n";
    delete [] name;  // memory freed
    name = getname();  // reuse freed memory

    cout << name << " at " << (int *) name << "\n";
    delete [] name;  // memory freed again
    return 0;
}

char * getname()  // return pointer to new string
{
    char temp[80];  // temporary storage
    cout << "Enter last name: ";
    cin >> temp;
    char * pn = new char[strlen(temp) + 1];
    strcpy(pn, temp);  // copy string into smaller space
    return pn;  // temp lost when function ends
}

Here is a sample run:

Enter last name: Fredeldumpkin
Fredeldumpkin at 0x004326b8
Enter last name: Pook
Pook at 0x004301c8

Program Notes

First, consider the function getname(). It uses cin to place an input word into the temp array. Next, it uses new to allocate new memory to hold the word. Including the null character, the program needs strlen(temp) + 1 characters to store the string, so that's the value given to new. After the space becomes available, getname() uses the standard