Programming Fundamentals

Pointers

Pointers

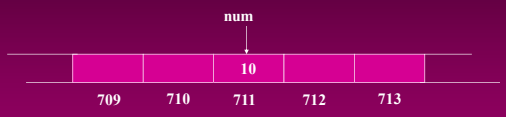
- One of the most powerful tools available to a C++ programmer is the ability to manipulate computer memory directly by using pointers

What Is a Pointer?

- A *pointer* is a variable that stores a memory address
- Computer memory is divided into sequentially numbered memory locations
- Each variable is located at a unique location in memory, known as its address
- The address stored in a pointer usually corresponds to the position in memory where a variable is located

What Is a Pointer?

int num = 10;



SJ2

Address-Of (&) Operator

- Before looking into Pointers, lets look at ***"address-of"*** operator (&)
- If we use address-of operator before a variable name it returns the memory address of that variable

```
int num = 10;
cout<<"Address:"<<&num<<endl;
```


Storing the Address in a Pointer

- Every variable has an address
- We can store that address in a pointer
- For declaring a pointer we use the character asterisk (*) before a variable name with the type of the address location which will be saved in it
- For example: if pNum is a pointer to an integer in memory then

```
int *pNum;
```

KSJ2 Example: project "address of op"

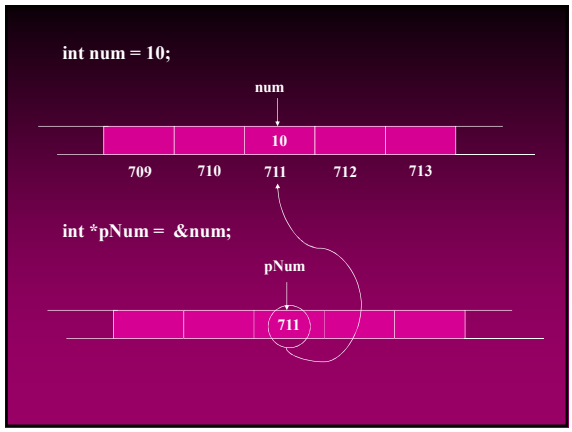
Ksjanjua, 12/2/2004

Initializing a Pointer

- A pointer whose value is zero is called a null pointer

  int *pNum = 0;

- All pointers when they are created, should be initialized to something. If you don't know what you want to assign to the pointer, assign 0
- A pointer that is not initialized is called a *"**wild pointer**"*
- Wild pointers are very dangerous



The Indirection Operator

- The indirection operator (*) is also called the dereference operator
- We may call it "value at this address-operator"
- When a pointer is dereferenced, the value at the address stored by the pointer is retrieved

```
cout<<"Value where pointing:" <<*pNum<<endl;
*pNum = 20;
```


The Indirection Operator

- From this example we can see that the indirection Operator means

  "the value stored at this address"

- Keep in mind that symbol * is used in two distinct ways with pointers: ***declaration*** and ***dereference***

The Indirection Operator

- When a pointer is declared, the asterisk (*) indicates that it is a pointer, not a normal variable

```
int *pNum;
```

- When the pointer is dereferenced, the indirection operator indicates the value at the memory location stored in the pointer

```
cout<<*pNum;
```


Pointers, Addresses, and Variables

- It is important to distinguish between a **pointer**, the **address** that the pointer holds, and the **value** at the address held by the pointer
- This is the source of much of the confusion about pointers
- Consider the next example

KSJ1 Example: project "pointer01"

Ksjanjua, 12/2/2004

Pointers, Addresses, and Variables

```
int theVariable = 5;
int * pPointer = &theVariable;
```

- TheVariable is declared to be an integer variable initialized with the value 5
- pPointer is declared to be a pointer to an integer; it is initialized with the address of theVariable
- The address that pPointer holds is the address of theVariable
- The value at the address that pPointer holds is 5