Programming Fundamentals (COMP1112)
Pointers

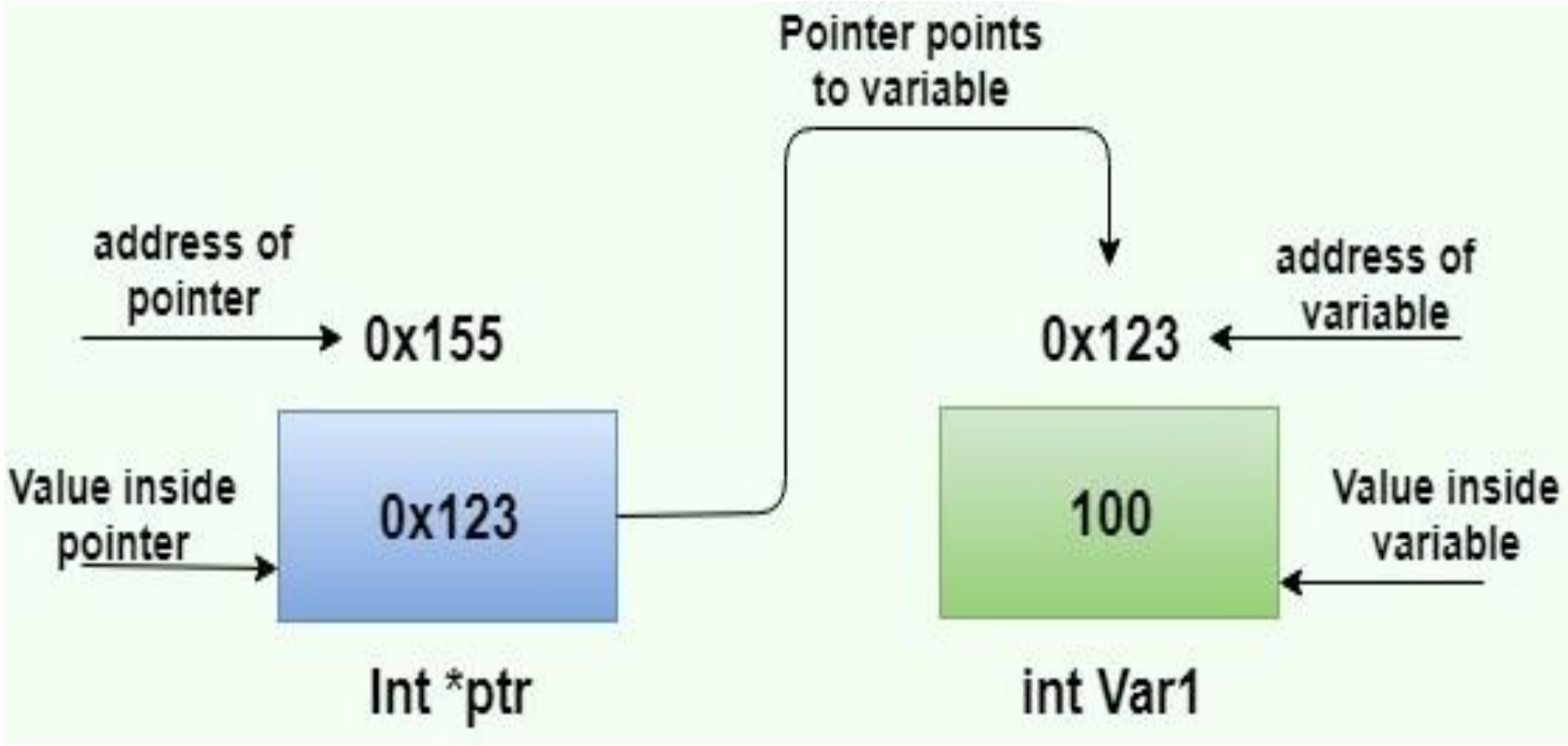
Pointer

- A pointer is a variable that is used to store a memory address. The reference operator is used to store the memory address of a variable and store it in a pointer.
- Pointer is declared as:

DataType *var;

- DataType : It is the type of variable pointed by the pointer variable
- * : It indicates that the var is a pointer variable
- var: It is name of the pointer variable

Pointer in C++


Pointer points
to variable
address of
pointer
0x155
0x123
address of
variable
Value inside
pointer
0x123
100
Value inside
variable
Int *ptr
int Var1

Example to use pointer

```
int n;
int *ptr;
cout<<“enter an integer”;
cin>>n;
ptr=&n;
cout<<“value of n:”<<n<<endl;
cout<<“address of n:”<<ptr<<endl;
```

Dereference operator

- Used to access the value of variable whose address is stored in pointer
- Denoted by *
- Also known as indirection operator

Pointer initialization

- The process of assigning a memory address to a pointer at the time of declaration is called pointer initialization.
- Pointer can be initialized to any valid memory address
- It can also be initialized to NULL or 0

DataType *p= &variable;

- DataType : It is the type of variable pointed by the pointer variable
- * : It indicates that the variable is a pointer variable
- &: address operator to access memory address of a variable
- variable: It is name of the pointer variable

Example: use of deference operator

```
int a, b, s, *p1, *p2;
p1=&a;
p2=&b;
cout<<“enter an integer”;
cin>>*p1;
cout<<“enter another integer”;
cin>>*p2;
s=*p1+*p2;
cout<<s;
```

Pointer addition/subtraction

- Addition/subtraction operator on pointer is used to move the reference forward/backward in memory
- Change of memory address depends on the data type of pointer

Pointer and arrays

- All elements of arrays are stored in consecutive memory locations.
- A pointer can access all elements of array if the address of first element is assigned to it
- The name of the array represents the address of it’s first element
- The address of first element can be assigned to a pointer by assigning the name of the array to pointer

```
int NUM[10];
int *ptr;
ptr=NUM;
```

Example: accessing array elements

```
int Num[5]={10, 20, 30, 40, 50};
int *ptr=Num;
cout<<*ptr;
ptr++;
cout<<*ptr;
```

- Array elements can also be accessed as

```
cout<<*ptr;
cout<<*(ptr+1);
```

Take input of five integers in an array and display them using pointer

```
int marks[5];
int i, *ptr;
cout<<“enter five marks”;
for (i=0;i<5;i++)
cin>>marks[i];
ptr=marks;
cout<<“you entered”;
for (i=0;i<5;i++)
cout<<*ptr++;
```

Example 1: Array of characters

```
char name[10];
cout<<"enter name";
cin>>name;
int i=0;
while (name[i]!='\0'){

cout<<name [i];
i++;
}
cout<<i;
```

Example 2: Array of characters

```
char name[20], *ptr;
cout<<“enter your name”;
cin.get(name, 20);
ptr=name;
cout<<“name entered is”<< ptr; //why not *ptr, it will print just first char
```

Note: the pointer displays the values stored in each element of the array name until it finds null character \0

References

- C++ How to Program
  By Deitel & Deitel

- The C++ Programming Language
  By Bjarne Stroustrup

- Object oriented programming using C++ by Tasleem Mustafa, Imran Saeed, Tariq Mehmood, Ahsan Raza

- https://www.tutorialspoint.com/cplusplus

- http://ecomputernotes.com/cpp/introduction-to-oop

- http://www.cplusplus.com/doc/tutorial

- https://www.guru99.com/c-loop-statement.html

- www.w3schools.com