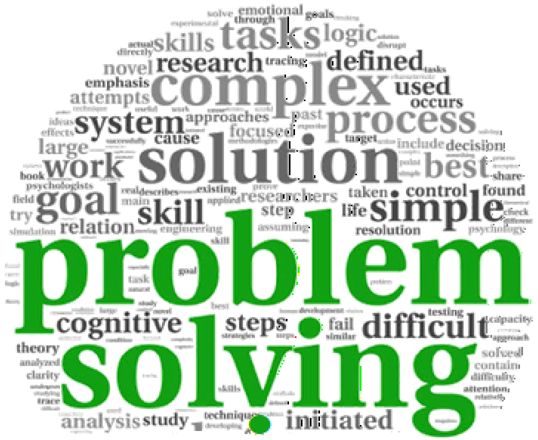
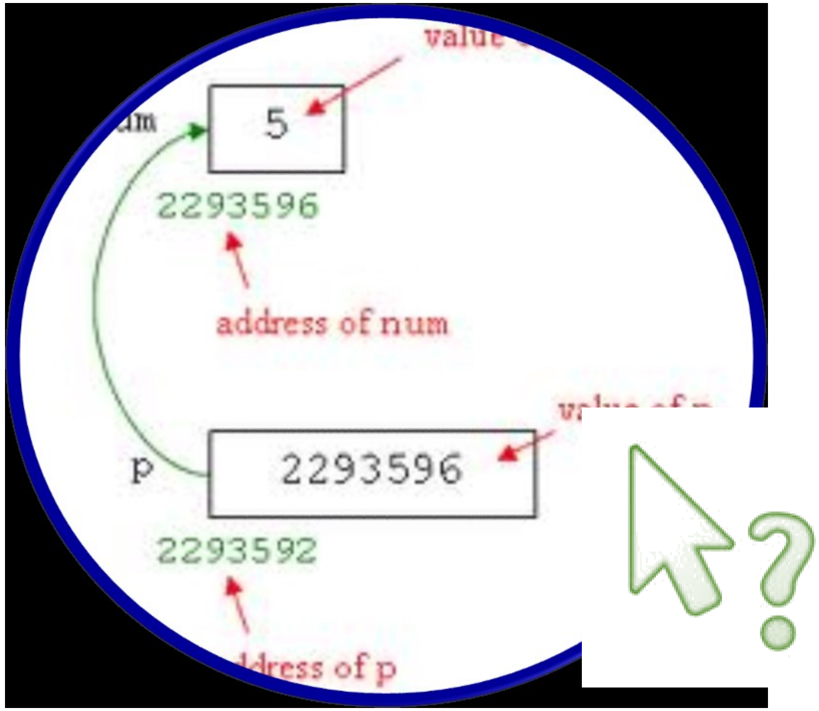


problem solving using computers

CSE 1051



Pointers

S16-1

Objectives

To learn and appreciate the following concepts:

- Concept of Basic Pointers – declaration and initialization
- Accessing the variable using address-of operator

Session outcome

At the end of session one will be able to:

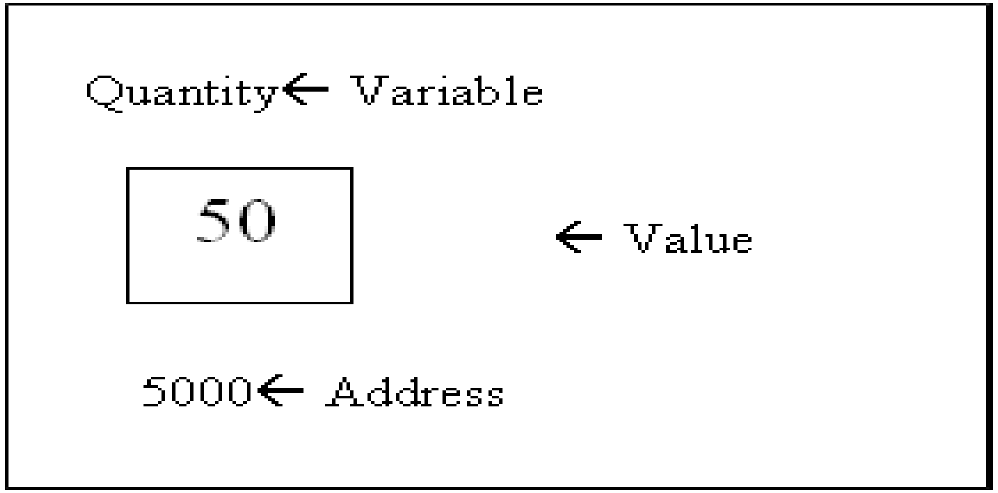
- Understand the concept of Basic Pointers
- Access the variable using address-of operator

Variable - Concept

- Consider the following statement

```
int Quantity =50;
```

- Compiler will allocate a memory location for Quantity and places the value in that location. Suppose the address of that location is 5000, then



Pointer

A memory location or a variable which stores the address of another variable in memory.

Commonly used in C than in many other languages (such as BASIC, Pascal, and certainly Java, which has no pointers).

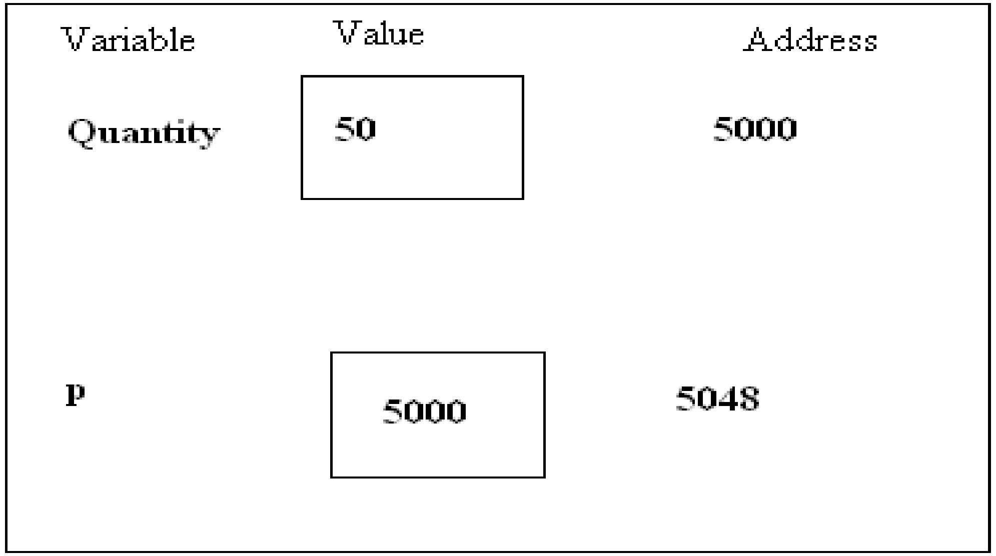
Pointer Concept

int Quantity=50 ;

- To assign the address 5000 (the location of quantity) to a variable **p**, we can write:

int *p = &Quantity ;

Such variables that hold memory addresses are called Pointer Variables.



Variable	Value	Address
Quantity	50	5000
p	5000	5048

Declaring and initializing pointers

Syntax:

data_type * pt_name;

This tells the compiler 3 things about the **pt_name:**

- The **asterisk**(*) tells the variable **pt_name** is a **pointer variable**.
- **pt_name** needs a **memory location**.
- **pt_name** points to a variable of type **data_ type**

Reference and dereference operators

& is the ‘reference’ operator and can be read as "address of"

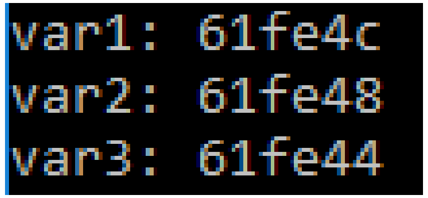
***** is the ‘dereference’ operator and can be read as “value at address” or "value pointed by"

Program to illustrate the address of operator

```
#include <stdio.h>
int main() {
 int var1 = 11;
 int var2 = 22;
 int var3 = 33;
//print the addresses of these variables
   printf(“%x”,&var1);
   printf(“%x”,&var2);
   printf(“%x”,&var3);
 return 0;
}
```

Output:

```
0x61fe4c
0x61fe48
0x61fe44
```



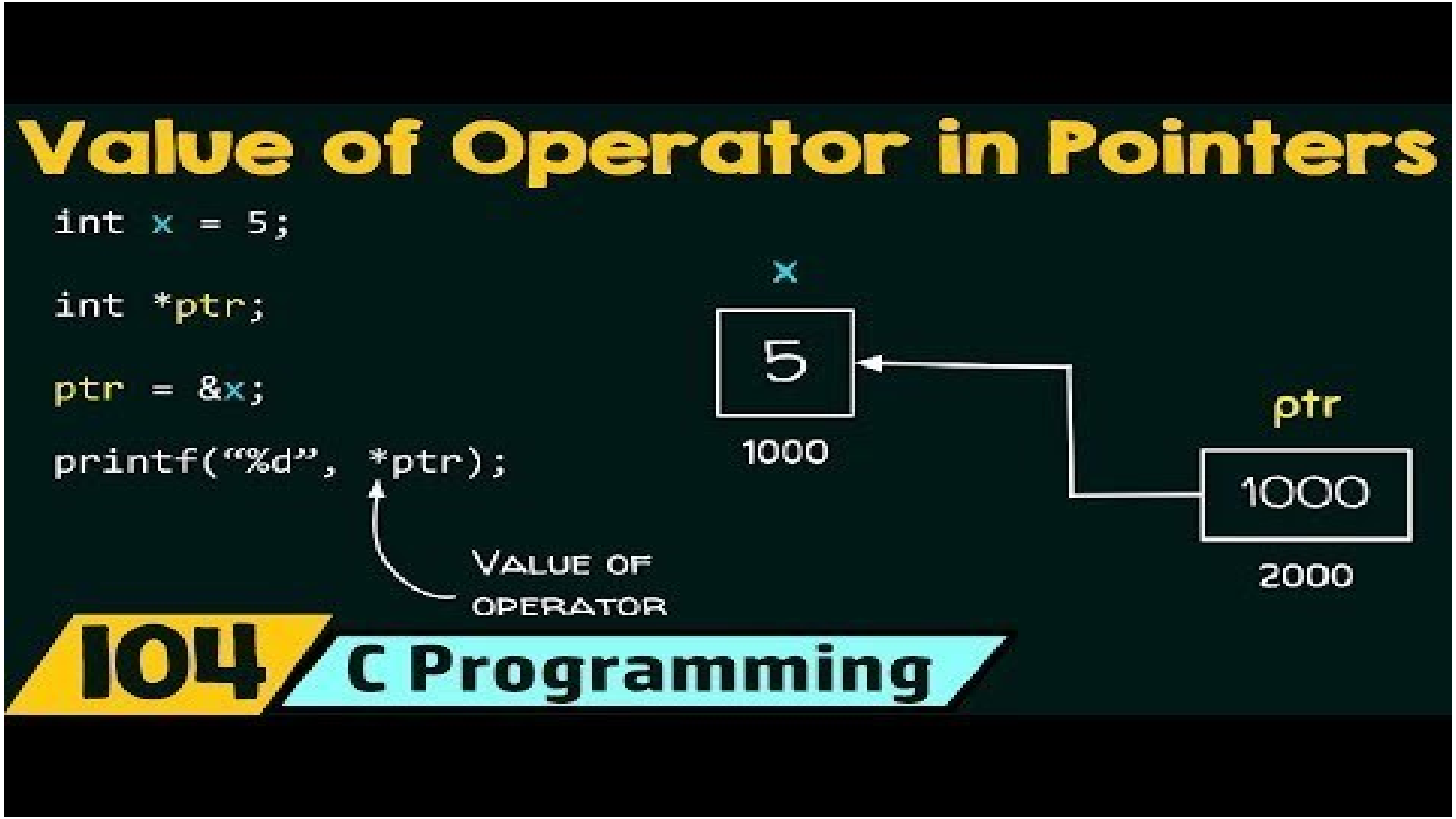
```
var1: 61fe4c
var2: 61fe48
var3: 61fe44
```

Go to posts/chat box for the link to the question

submit your solution in next 2 minutes

The session will resume in 3 minutes

Understanding pointers better



```
int x = 5;

int *ptr;

ptr = &x;

printf(“%d”, *ptr);
```

https://www.youtube.com/watch?v=xlt_bEqfnxg

Summary till now …

```
int  v;      //defines variable v of type int
int* p;      //defines p as a pointer to int

p = &v;      //assigns address of variable v to  pointer p
```

Now…

```
  v = 3;     //assigns 3 to v
```

✓ **&** is the ‘reference’ operator and can be read as "address of"

✓ ***** is the ‘dereference’ operator and can be read as “value at address” or "value pointed by"

To be taken care …

- Before a pointer is initialized, it should not be used.
- We must ensure that the pointer variables always point to the corresponding type of data.
- Assigning an absolute address to a pointer variable is prohibited. i.e p=5000
- A pointer variable can be initialized in its declaration itself.

Example:

```
int x, *p=&x;     //declares x as an integer
                    variable and then initializes
                    p to the address of x.
```

To be taken care …

The statement

int *p = & x, x;   not valid.

i.e target variable ‘x’ must be declared first.

Summary

- Concept of Basic Pointers – declaration and initialization
- Accessing the address of a variable using & operator