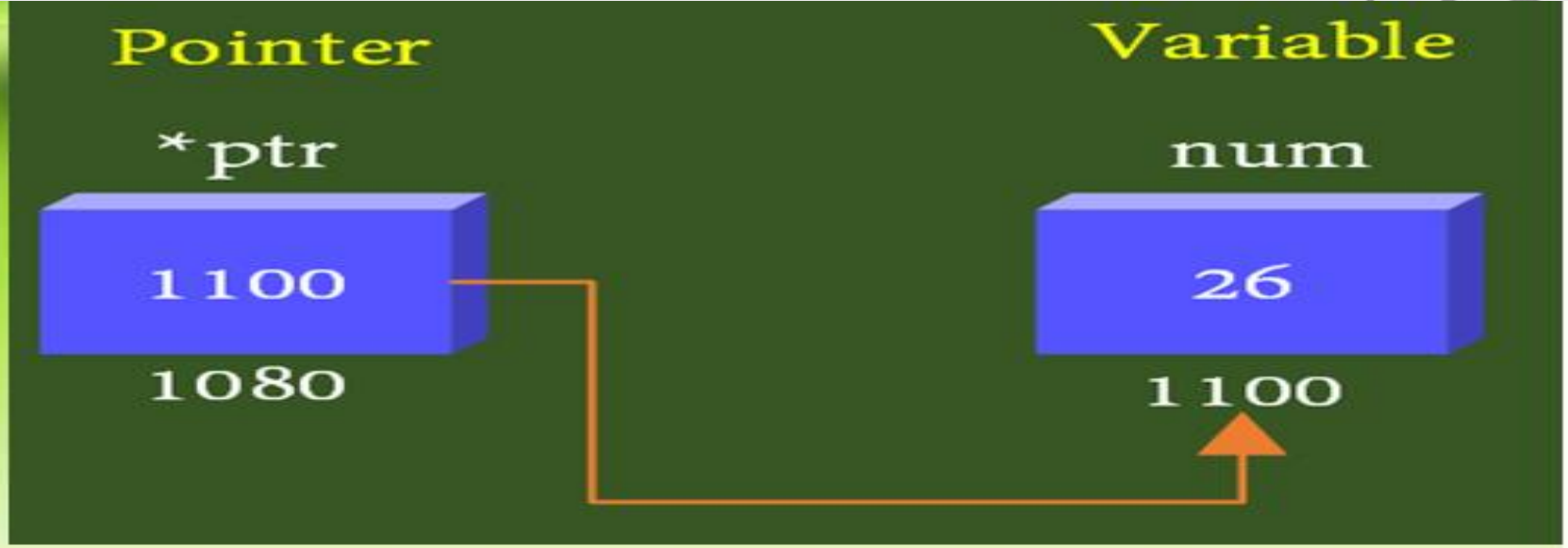
UNIVERSITY of SAN CARLOS
SCIENTIA • VIRTUS • DEVOTIO

CIS 1101 – PROGRAMMING 1

POINTERS IN C



POINTERS

POINTER: DEFINITION

- A variable that stores or points the address of another variable

- A variable whose value is the address of another variable

- Used to allocate memory dynamically

POINTER: CONSTANTS

- Addresses in memory or the set of addresses for a computer.

- Exist by themselves and cannot be changed.

- Can only be used by naming them with variables.

ADDRESS EXPRESSION

- **Syntax:**

  &variable_name

- **Note:**
  - The "**address of" operator** (**&**) extracts the **address for a variable**.

POINTER VARIABLE

- A variable that **stores** a pointer

- A variable that is used to store the **address** of another variable

- **Examples:**

  p, q, ptr, and others

POINTER: IMPORTANT NOTES

- The pointer variable might be belonging to any of the **data type** such as:
  - ✓**int**
  - ✓**float**
  - ✓**char**
  - ✓**double**
  - ✓**short**
  - ✓and **more**

- A pointer can be used to point to any variable type.

POINTER DECLARATION: SYNTAX

```
datatype* var_name;
```

- **Example:**

```
int* ptr;
```

POINTER DECLARATION: SYNTAX

- **Example:**

int* ptr;

- A pointer "**ptr**" holds the address of an **integer variable.**

- The operator "*****" is used to denote that "**ptr**" is a pointer variable and not a normal variable.

POINTER: OPERATORS

- **Address-of Operator (&):**
  - A unary or monadic operator that gives the address of a variable.
  - Used to get the address of the variable.

POINTER: OPERATORS

- **Indirection or Dereference operator (*):**
  - A unary or monadic operator that gives the **contents of an object** pointed to by a pointer.
  - Used to get the value of the variable that the pointer is pointing to

USES OF * OPERATOR

1) Declaration:

- Used to declare a pointer variable in C

  - **Examples:**

```
int* pa;
int* pb;
```

USES OF * OPERATOR

2) Redirection:

- Used to access the value stored in the address

- Returns the value of the variable
- located at the address specified by its operand

- **Redirects** the operation
- from the **pointer variable** to a **data variable**

- **Example:**

sum = *pa + *pb;

USING A POINTER

```
#include <stdio.h>

int main()
{
    /* A normal integer variable Var */
    int Var = 10;

    /* A pointer variable that holds address of Var.  */
    int* pvar = &Var;

    /* This line prints value at address stored in Var */
    /* The value stored is the value of variable “Var" */
    printf("The value  of  Var = %d\n", *pvar);
```

USING A POINTER: CONTINUATION

```
/* The output of this line may be different in different runs even on same machine. */
    printf("\nThe address of  Var = %p\n", pvar);


/* Use pvar as lvalue (Left hand side of assignment) */
    *pvar = 20;   /* Value at address is now 20 */


/* This line prints 20 */
    printf("\nAfter doing *pvar = 20, *pvar is %d\n", *pvar);

    return 0;
}
```

REFERENCING

- Means taking the address of an existing variable (using **&**) to set a pointer variable.

- **Example:**

```
int c;
int* pc;
c = 5;
pc = &c;
```

- **Note:** **pc references c**

DEREFERENCING

- Using the * operator (asterisk character) to retrieve the value from the memory address that is pointed by the pointer.

- **Note:**

  - The value stored at the address of the pointer must be a value **OF THE SAME TYPE** as the **type of variable** the pointer "points" to.

- **Example:**

```
int n;
n = *pn;
```