

# Programming with C I

Fangtian Zhong  
CSCI 112

Gianforte School of Computing  
Norm Asbjornson College of Engineering  
E-mail: [fangtian.zhong@montana.edu](mailto:fangtian.zhong@montana.edu)

# Pointers

## ➤ pointer (pointer variable)

- A pointer is a variable whose value is the address of a location in memory.
- 8 bytes on server but depends on machine
- syntax: *type \*variable*

```
int m = 25;  
int *itemp=&m;    /* a pointer to an integer */  
char q= 'c';  
char *ch = &q; /*a pointer to a character*/
```

## **& operator (address of)**

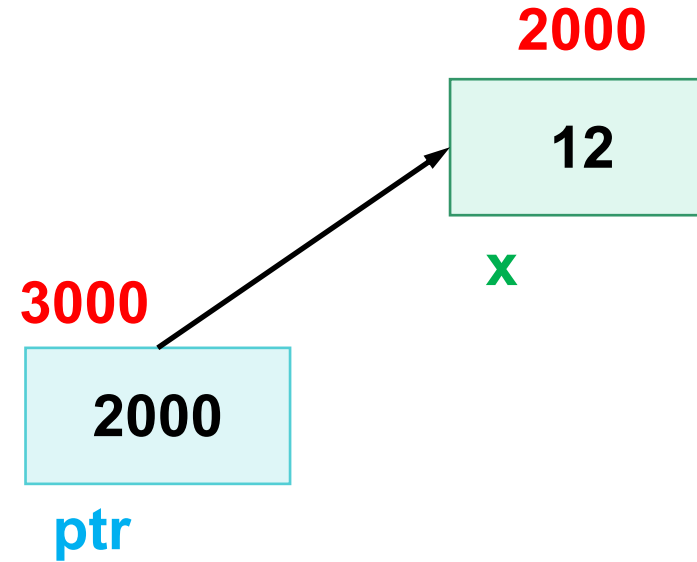
**➤ Returns the address of a variable**

**the \* *never* returns the address of a variable**

# Using a Pointer Variable

```
int x;  
x=12;
```

```
int* ptr;  
ptr=&x;
```

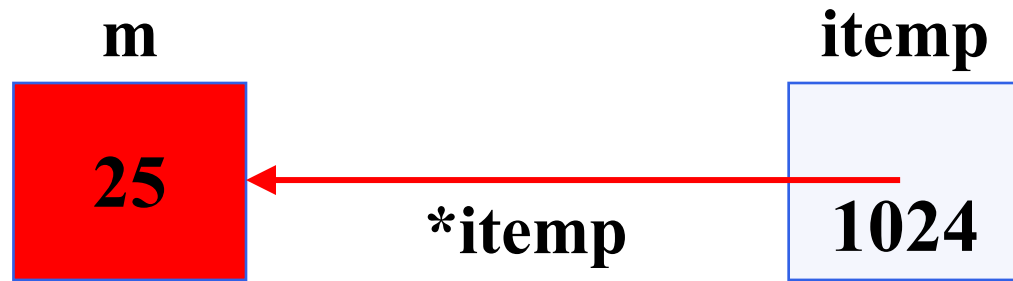


**NOTE:** Because ptr holds the address of x, we say that ptr “points to” x.



# Indirection/indirect reference

- accessing the contents of a memory cell through a pointer variable that stores its address



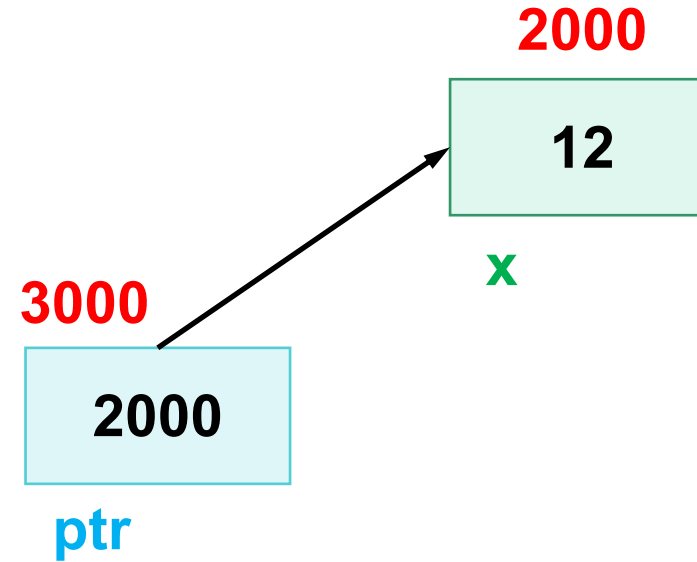
➤ **Figure Referencing a Variable Through a Pointer**

**Table** References with Pointers

Reference	Cell Referenced	Cell Type Value)
itemp	gray shaded cell	pointer (1024)
*itemp	cell in color	int (25)

# \*: dereference operator

```
int x;  
x=12;  
  
int* ptr;  
ptr=&x;  
printf(“%d\n”,*ptr);
```

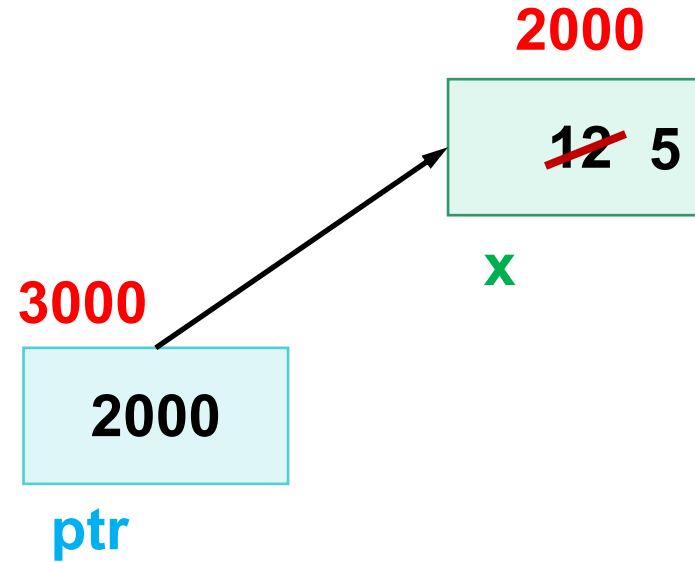


**NOTE:** The value pointed to by ptr is denoted by \*ptr.



# Using the Dereference Operator

```
int x;  
x=12;  
  
int* ptr;  
ptr=&x;  
*ptr=5;
```



// changes the value at the address ptr points to 5



## **\* operator (indirection)**

- **Follows a pointer to what it points to**
- **(the thing at the address it stores)**



# Pointers to Files

- **C allows a program to explicitly name a file for input or output.**
- **Declare file pointers:**
  - `FILE *inp; /* pointer to input file */`
  - `FILE *outp; /* pointer to output file */`
- **Prepare for input or output before permitting access:**
  - `inp = fopen("infile.txt", "r");`
  - `outp = fopen("outfile.txt", "w");`

# Pointers to Files

## ➤ **fscanf**

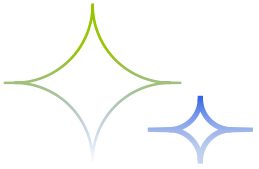
- file equivalent of scanf
- `fscanf(inp, "%lf", &item);`

## ➤ **fprintf**

- file equivalent of printf
- `fprintf(outp, "%.2f\n", item);`

## ➤ **closing a file when done**

- `fclose(inp);`
- `fclose(outp);`



# THE END

Fangtian Zhong  
CSCI 112

Gianforte School of Computing  
Norm Asbjornson College of Engineering  
E-mail: [fangtian.zhong@montana.edu](mailto:fangtian.zhong@montana.edu)