

# **Introduction to Scientific and Engineering Computation (BIL 104E)**

## **Lab 8**

# Pointers: The Address-of Operator (&)

## Obtaining the Left Values of Variables

```
1:  /* 11L01.c: Obtaining addresses */
2:  #include <stdio.h>
3:
4:  main()
5:  {
6:      char  c;
7:      int   x;
8:      float y;
9:
10:     printf("c: address=%p, content=%c\n", &c, c);
11:     printf("x: address=%p, content=%d\n", &x, x);
12:     printf("y: address=%p, content=%5.2f\n", &y, y);
13:     c = 'A';
14:     x = 7;
15:     y = 123.45;
16:     printf("c: address=%p, content=%c\n", &c, c);
17:     printf("x: address=%p, content=%d\n", &x, x);
18:     printf("y: address=%p, content=%5.2f\n", &y, y);
19:     return 0;
20: }
```

# Pointers: The Address-of Operator (&)

## Computer Screen

```
c: address=0022FF47, content=v  
x: address=0022FF40, content=2293576  
y: address=0022FF3C, content= 0.00  
c: address=0022FF47, content=A  
x: address=0022FF40, content=7  
y: address=0022FF3C, content=123.45
```

# Pointers: Declaring Pointers

## Declaring and Assigning Values to Pointers

```
1:  /* 11L02.c: Declaring and assign values to pointers */
2:  #include <stdio.h>
3:
4:  main()
5:  {
6:      char  c, *ptr_c;
7:      int   x, *ptr_x;
8:      float y, *ptr_y;
9:
10:     c = 'A';
11:     x = 7;
12:     y = 123.45;
13:     printf("c: address=%p, content=%c\n", &c, c);
14:     printf("x: address=%p, content=%d\n", &x, x);
15:     printf("y: address=%p, content=%5.2f\n", &y, y);
16:     ptr_c = &c;
17:         printf("ptr_c: address=%p, content=%p\n", &ptr_c, ptr_c);
18:         printf("*ptr_c => %c\n", *ptr_c);
19:     ptr_x = &x;
20:         printf("ptr_x: address=%p, content=%p\n", &ptr_x, ptr_x);
21:         printf("*ptr_x => %d\n", *ptr_x);
22:     ptr_y = &y;
23:         printf("ptr_y: address=%p, content=%p\n", &ptr_y, ptr_y);
24:         printf("*ptr_y => %5.2f\n", *ptr_y);
25:     return 0;
26: }
```

# Pointers: Declaring Pointers

## Computer Screen

```
c: address=0x1B38, content=A
x: address=0x1B36, content=7
y: address=0x1B32, content=123.45
ptr_c: address=0x1B30, content=0x1B38
*ptr_c => A
ptr_x: address=0x1B2E, content=0x1B36
*ptr_x => 7
ptr_y: address=0x1B2C, content=0x1B32
*ptr_y => 123.45
```

# Pointers: Updating Variables via Pointers

## Changing Variable Values Via Pointers

```
1:  /* 11L03.c: Changing values via pointers */
2:  #include <stdio.h>
3:
4:  main()
5:  {
6:      char  c,  *ptr_c;
7:
8:      c = 'A';
9:      printf("c: address=%p, content=%c\n", &c, c);
10:     ptr_c = &c;
11:     printf("ptr_c: address=%p, content=%p\n", &ptr_c, ptr_c);
12:     printf("*ptr_c => %c\n", *ptr_c);
13:     *ptr_c = 'B';
14:     printf("ptr_c: address=%p, content=%p\n", &ptr_c, ptr_c);
15:     printf("*ptr_c => %c\n", *ptr_c);
16:     printf("c: address=%p, content=%c\n", &c, c);
17:     return 0;
18: }
```

# Pointers: Updating Variables via Pointers

## Computer Screen

```
c: address=0x1828, content=A  
ptr_c: address=0x1826, content=0x1828  
*ptr_c => A  
ptr_c: address=0x1826, content=0x1828  
*ptr_c => B  
c: address=0x1828, content=B
```

# Pointers: Pointing to the Same Memory Location

## Pointing to the Same Memory Location with More Than One Pointer

```
1:  /* 11L04.c: Pointing to the same thing */
2:  #include <stdio.h>
3:
4:  main()
5:  {
6:      int x;
7:      int *ptr_1, *ptr_2, *ptr_3;
8:
9:      x = 1234;
10:     printf("x: address=%p, content=%d\n", &x, x);
11:     ptr_1 = &x;
12:     printf("ptr_1: address=%p, content=%p\n", &ptr_1, ptr_1);
13:     printf("*ptr_1 => %d\n", *ptr_1);
14:     ptr_2 = &x;
15:     printf("ptr_2: address=%p, content=%p\n", &ptr_2, ptr_2);
16:     printf("*ptr_2 => %d\n", *ptr_2);
17:     ptr_3 = ptr_1;
18:     printf("ptr_3: address=%p, content=%p\n", &ptr_3, ptr_3);
19:     printf("*ptr_3 => %d\n", *ptr_3);
20:     return 0;
21: }
```



# Pointers: Pointing to the Same Memory Location

## Computer Screen

```
x: address=0x1838, content=1234  
ptr_1: address=0x1834, content=0x1838  
*ptr_1 => 1234  
ptr_2: address=0x1836, content=0x1838  
*ptr_2 => 1234  
ptr_3: address=0x1832, content=0x1838  
*ptr_3 => 1234
```