CSE 113 Structured Programming Language

# Pointer

Tasnim Zahan Assistant Professor Dept. of CSE NEUB

#### Pointers

- A pointer is a variable that stores the address of another variable
- For example, an integer variable holds an integer value, however an integer pointer holds the address of an integer variable

#### Address of a variable

```
#include <stdio.h>
int main()
{
   int num = 10;

   printf("Value of variable num is: %d", num);
   printf("\nAddress of variable num is: %p", &num);

   return 0;
}
```



#### Output:

Value of variable num is: 10
Address of variable num is: 0060FEFC

#### Pointer declaration

The data type of the pointer variable and the variable that it points must match

```
int *p1  //Pointer to an integer variable
double *p2  //Pointer to a variable of data type double
char *p3  //Pointer to a character variable
float *p4  //pointer to a float variable
```

### Pointer declaration

Address of variable num is: 0060FEF8

```
#include <stdio.h>
int main()
  int num = 10;
  //Pointer declaration
  int *p;
  //Assigning address of num to the pointer p
  p = #
  printf("Address of variable num is: %X", p);
  return 0;
Output:
```

# Access variable using pointer

```
double data = 10;
double *p;
p = &data;
```

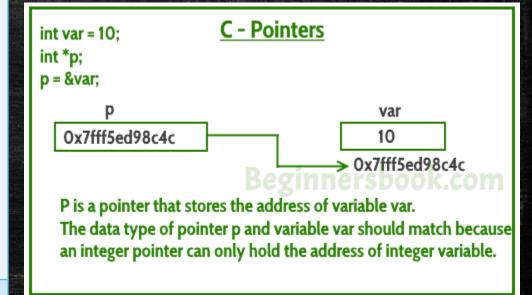
- using \* operator we can access the value of a variable through a pointer
- \*p give us the value of the variable data

```
printf("%d", *p);
printf("%d", data);
```

We can change value of data using \* operator

```
*p = 200;
printf("%d", data);
```

```
int main()
   int *p;
   int var = 10;
   p= &var;
   printf("Value of the var: %d", var);
   printf("\nValue of the var: %d", *p);
   printf("\nAddress of the var: %X", &var);
   printf("\nAddress of the var: %X", p);
   return 0;
Output:
Value of var is: 10
Value of var is: 10
Address of var is: 0x7fff5ed98c4c
Address of var is: 0x7fff5ed98c4c
```



# Find the output

```
2.
1. int *ptr, c;
                                      int *ptr,
    c = 5;
                                      int c;
                                      c = 5;
    ptr = &c;
    c = 20;
                                      ptr = &c;
    printf("%d", c);
                                      printf("%d", *ptr);
    printf("%d", *ptr);
                                      *ptr = 40;
                                      printf("%d", c);
3. | int* ptr, x, y;
                                  4. int* ptr, x, y;
    x = 5;
                                      x = 5;
    y = -15;
                                      y = -15;
    ptr = &x;
                                      ptr = &x;
    printf("%d", *ptr);
                                      printf("%d", *ptr);
                                      *ptr = 100
    ptr = &y;
    printf("%d", *ptr);
                                      ptr = &y;
                                      printf("x=%d, y=%d", x,y);
```

# operators & and \*

- "Address of"(&) Operator
- Using (&) operator we can get the address of a variable
- printf("Address of var is: %X", &num);
- "Value at Address"(\*) Operator
- Using \* operator we can access the value of a variable through a pointer
- printf("Value of the variable is: %d", \*ptr);

## Pointer to function

```
void pointer func (int a, int b, int c, int *L, int *S)
{
    if(a>b && a>c) *L=a;
    else if(b>a && b>c) *L=b;
    else *L=c;
    if(a<b && a<c) *S=a;
    else if(b<a && b<c) *S=b;
    else *S=c;
int main()
    int x,y,z, lar, sml;
    printf("Enter 3 numbers:\n");
    scanf("%d %d %d", &x,&y,&z);
    pointer func(x,y,z, &lar, &sml);
    printf("Largest: %d, smallest: %d", lar,sml);
    return 0;
```

# Swapping two numbers using Pointers

```
#include <stdio.h>
void swapnum(int *num1, int *num2);
int main()
   int v1 = 11, v2 = 77;
   printf("Before swapping:");
   printf("\nValue of v1 is: %d", v1);
   printf("\nValue of v2 is: %d", v2);
   /*calling swap function*/
   swapnum( &v1, &v2 );
   printf("\nAfter swapping:");
   printf("\nValue of v1 is: %d", v1);
   printf("\nValue of v2 is: %d", v2);
```

```
void swapnum(int *num1, int *num2)
{
  int tempnum;

  tempnum = *num1;
  *num1 = *num2;
  *num2 = tempnum;
}
```