

CSCI 2132

Software Development

Lecture 21:

Pointers

Instructor: Vlado Keselj

Faculty of Computer Science

Dalhousie University

Previous Lecture

- Program organization
- External (global) variables
- Pointers: History

Pointers

- Very important features of C
- Harold Lawson credited with invention of pointers in 1964, when he introduced them in PL/I programming language
- Use in Pascal (1970), but more restricted
- Pointers allow us to refer to a memory address
- Computers generally assign a unique integer, starting from 0 to each byte in memory

Pointer Variables

- **Pointer variable** stores a memory address

- Example of declaration:

```
type *pointer_name;
```

- Example:

```
int *p;
```

```
char *p;
```

```
int **r;
```

- *Reference type* of a pointer

Address Operator

- *Address operator: &*
- Example:

```
int i, *p;  
p = &i;
```

- or:

```
int i;  
int *p = &i;
```

Indirection Operator

- *Indirection operator: **
- used to access value of the object pointed by a pointer
- Example:

```
int i = 7;  
int *p = &i;  
printf("%d\n", *p);
```

What is the Code Output?

```
1: int i = 7;  
2: int *p = &i;  
3: (*p)++;  
4: printf("%d %d\n", i, *p);
```

Common Bugs with Pointers

1. Dereferencing an un-initialized pointer

- Result: undefined behaviour
- Example: `int *p; *p = 5;`

2. Dangling pointer

- Accessing an object that does not exist any more on stack or heap
- Example:

```
int* f() { int i=4; return &i; }  
...  
int *p; p = f(); ++(*p);
```


Pointer Assignment

- Use assignment operator: =
- Must be pointers of the same type
- Example:

```
int i = 8, j = 15;  
int *p = &i;  
int *q;  
int *r = &j;
```

```
*r = *p;  
q = p;  
(*q)++;
```

```
printf("%d %d %d %d %d\n", i, j, *p, *q, *r);
```

Pointer Arguments

- Allow to modify variables in caller function
- Remember the use of `scanf` function
- Example:

```
void swap(int *a, int *b) {  
    int temp = *a;  
    *a = *b;  
    *b = temp;  
}
```

```
int a = 4;  
int b = 5;  
swap(&a, &b);  
printf("%d %d\n", a, b);
```

Example: statistics.c

- Let us look at the fill-in-the-blanks code of statistics.c, available at:

```
~prof2132/public/statistics.c  
on bluenose
```

- Compilation using math library:

```
gcc -lm statistics.c
```