

# Types, Operators and Expressions

Nalini Vasudevan  
Columbia University

# Constants

```
#include<stdio.h>
#define PI 3.14  /* PI is a constant */
main ()
{
    int r;
    float area;
    scanf ("%d", &r);
    area = PI * r * r;
    printf ("Area = %d", area);
}
```

# Operators

- Arithmetic operators

- +, -, \*, /, %

- + and - have a lower precedence than \*, /, %

Swapping two numbers

# Swapping two numbers

```
int temp;
```

```
temp = x;
```

```
x = y;
```

```
y = temp;
```

# Puzzle

- How do you swap two numbers without using a temp variable?
  - Hint: Use arithmetic operators + and -

# Relational Operators

- `>` , `>=` , `<` , `<=` , `==` , `!=`
- Example

```
if (a < b)
    printf ("a is less than b\n")
else
    printf ("b is less than or equal to a\n");
```

# Logical Operators

- &&, ||
- Example

```
if (a == 5 || b == 5 )  
    printf ("There is at least one 5\n")  
else  
    printf ("There isn't any 5\n");
```

# Increment and Decrement Operators

- ++ , --

- Prefix increment

```
c = 5;
```

```
x = ++c; /* value of c is 6 and x  
is 6 */
```

# Increment and Decrement Operators

- Postfix increment

```
c = 5;
```

```
x = c++;
```

```
/* value of c is 6 but x is 5 */
```

– Equivalent to:

```
x = c;
```

```
c = c + 1;
```

# Bitwise operators

- Bitwise AND &
- Bitwise OR |
- Bitwise exclusive OR ^
- Left shift <<
- Right shift >>
- One's complement ~

# Puzzle

- How do you find if a number is a power of 2?

```
if (...)
```

```
    printf ("Power of 2\n");
```

```
else
```

```
    printf ("Not a power of 2\n");
```

- Hint

- Use bitwise operator &

# Puzzle

- How do you swap two numbers using bitwise operators
  - Hint: Use  $\wedge$

# Assignment Operators

```
i = i + 2;
```

is equivalent to

```
i += 2;
```

```
i = i * 2;
```

is equivalent to

```
i *= 2;
```

# Conditional expressions

```
if (a > b)
    z = a;
else
    z = b;
```

is equivalent to

```
z = (a > b) ? a : b;
```

# Arrays

- Declaration

```
int score[10];
```

- Reference

```
score[3] = 10;
```

- Index starts from 0 (goes till 9)
- Can use variables for indexing

```
i = 5;
```

```
score[i] = 2;
```

# Multi-dimensional arrays

- Array of arrays

```
int a[5][10];
```

```
a[2][3] = 9;
```

# Strings

- Array of characters

```
char str [20];
```

- Built in string functions

```
#include <string.h>
```

- `strcpy(str, "Nalini");`
- `i = strlen(str); /* i is 6 */`
- `'N', 'a', 'l', 'i', 'n', 'i', '\0'`

# Enumeration

```
#include<stdio.h>
enum days {SUN, MON, TUE, ... SAT};

main()
{
    enum days day;
    day = MON;
    if (day == SAT || day == SUN )
        printf ("Lets party!");
    else
        printf ("Lets try to work");
}
```

# Enumeration

```
#include<stdio.h>
enum days {SUN, MON, TUE, ... SAT};
/* SUN = 0, MON = 1 and so on.. */

main()
{
    enum days day;
    day = 1; /* Same as day = MON */
    if (day == SAT || day == SUN )
        printf ("Lets party!");
    else
        printf ("Lets try to work");
}
```

# Enumeration

```
#include<stdio.h>
enum days {SUN = 1, MON = 3, TUE, ... SAT};

main()
{
    enum days day;
    day = 4;
    if (day == SAT || day == SUN )
        printf ("Lets party!");
    else
        printf ("Lets try to work!");
}
```