Variables

* Allows us to refer to values by name
* Must be declared before first use
* Declaration must provide type, name, initial value (optional)
Examples:

```c
int a;       /* 2's complement integer */
unsigned int b; /* unsigned integer */
float volume, height; /* floating-point, 32 bits */
double pi, epsilon; /* floating-point, 64 bits */
char c = 'A'; /* ASCII representation of character */
```
* Variable scope defined by location of declaration:

1) Local: if defined inside some function
2) Global: defined outside of any function

* Example:

```c
#include <stdio.h>

float pi = 3.1415;    /* Global variable */

int main ()
{
    int i=1;           /* Local variable */
    printf("pi=%f, i=%d", pi, i);

    return 0;
}
```
Operators

* Assignment (=): value on the right will be assigned to variable on the left
  Data conversion will take place if necessary

```c
int i;
float f;
i = 2;       /* i holds value of 2 */
f = i;        /* f holds value of 2 now */
```
* Bitwise operators:

- Bitwise NOT: `~`
- Bitwise AND: `&`
- Bitwise OR: `|`
- Bitwise XOR: `^`
- Left-shift: `<<`
- Right-shift: `>>`

Examples:

- `y = ~x; /* x and y have bits flipped */`
- `z = y<<2; /* shift-left y two positions */`
- `w = (y&z)|x;`
* Arithmetic operators:
  
  Addition:    +    Division:  /  
  Subtraction: −    Remainder: %  
  Multiplication: *  

* Example:

  \[ z = (x+y)*(w\%m) ; \]

  \[ f = x+y*z+w; \quad /* \text{Be careful what you mean!} */ \]

  \[ g = (x+y)*(z+w); \quad /* \text{Use parenthesis to make clear evaluation order} */ \]
Control structures

* Sequential construct

Flow diagram

Statement 1

Statement 2

Statement 3

Example

```c
/* get equation coefficients */
printf("Enter a, b, and c: ");
scanf("%f %f %f", &a, &b, &c);
printf("Solving equation \%fx^2+\%fx+\%f=0\n", a, b, c);

/* compute solution */
D = b * b - 4 * a * c; /* compute determinant */
```
*Conditional construct*

Flow diagram

```
if (D == 0)
    x1 = -b / (2 * a);
```
Flow diagram

Test condition

Statement 1  Statement 2

Example

if (D > 0)
    printf("x1=%f, x2=%f\n", x1, x2);
else
    printf("No real roots exist\n");
Relational operators

Less: <
Less or equal: <=
Greater: >
Greater or equal: >=
Equal: ==
Not equal: !=

Example

if (x==0)
    printf("Zero");
else
    printf("Not zero");
Cascaded if-else and compound statements

```c
if (D > 0)
{
    x1 = (-b + sqrtf(D)) / (2 * a); /* Compound statement */
    x2 = (-b - sqrtf(D)) / (2 * a);
}
else if (D == 0)
    x1 = -b / (2 * a); /* Cascaded if */
/* Simple statement */
```
Example: solving \( ax^2 + bx + c = 0 \)

```c
#include <stdio.h>   /* needed for printf and scanf */
#include <math.h>    /* needed for sqrtf */

int main()
{
    float a, b, c;    /* quadratic equation coefficients */
    float D;          /* determinant */
    float x1, x2;     /* solution(s) */

    /* get equation coefficients */
    printf("Enter a, b, and c: ");
    scanf("%f %f %f", &a, &b, &c);
    printf("Solving equation \( ax^2 + bx + c = 0 \)\n", a, b, c);

    /* compute solution */
    D = b * b - 4 * a * c;    /* compute determinant */
    if (D > 0)                 /* two real roots exist */
    {
        x1 = (-b + sqrtf(D)) / (2 * a);
        x2 = (-b - sqrtf(D)) / (2 * a);
    }
    else if (D == 0)           /* only one root exists */
    
        x1 = -b / (2 * a);

    /* print results */
    if (D > 0)
        printf("x1=%f, x2=%f\n", x1, x2);
    else if (D == 0)
        printf("x=%f\n", x1);
    else
        printf("No real roots exist\n");
    return 0;
}
```