Data types

`int`: integer
`float`, `double`: decimal/fraction
`char`: text character

Variables

Declaring a variable:

Syntax: *datatype varname* ;
Examples:
```
int count;
double velocity;
```

Assigning a value to a variable:

Syntax: *varname* = *value* ;
Examples:
```
count = 0;
velocity = 9.81 * height;
```

Output with `printf`

Syntax:
```
printf( format );
printf( format, values );
```

Examples:
```
printf("Hello, world\n");
printf("Count is %i\n", count);
printf("Velocity is %.2lf m/s\n",
       velocity);
```

Input with `scanf`

Syntax:
```
scanf( format, & varname );
```

Examples:
```
scanf("%i", &count);
scanf("%lf", &velocity);
```

`printf`/`scanf` placeholders

`int`	`%i` or `%d`
`float`	`%f`
`double`	`%lf`
`char`	`%c`

`if/else` statements

```
if ( condition ) {
    statements
}

if ( condition ) {
    statements
} else {
    statements
}

if ( condition1 ) {
    statements
} else if ( condition2 ) {
    statements
} else {
    statements
}
```

"Keep going" loop

```
int keep_going = 1;
while (keep_going == 1) {
    statements
    if ( need to stop ) {
        keep_going = 0;
    }
}
```

Comparisons

Syntax: *value op value*
op is one of:
`==`, `!=` : equals, does not equal
`<`, `<=` : less than, less than or equal
`>`, `>=` : greater than, greater than or equal

Logic

Syntax: *condition op condition*
op is one of:
`||` : or, true if either condition is true
`&&` : and, true if both conditions are true

Loop recipes

Count from 1 to `n`:

```
for (int i = 1; i <= n; i++) {
  statements
}
```

Count from 0 to `n-1`:

```
for (int i = 0; i < n; i++) {
  statements
}
```

Count down from `n` to 1:

```
for (int i = n; i >= 1; i--) {
  statements
}
```

Count from 1 to `n` by increments:

```
for (int i = 1; i <= n; i += incr) {
  statements
}
```

Compute sum of `n` terms:

```
double sum = 0.0;
for (int i = 1; i <= n; i++) {
  double term = compute term i;
  sum += term;
}
```

Arithmetic

Syntax: *value op value*
op is one of:
`+` : addition (lower precedence)
`-` : subtraction (lower precedence)
`*` : multiplication (higher precedence)
`/` : division (higher precedence)
`%` : integer modulus (higher precedence)

Good luck!

...and don't panic!