

C Intro

Comments

```
// This is a line comment. Anything that comes after // is ignored by the compiler (gcc)

/*
   This is a block comment.
   Anything between the symbols will be ignored by the compiler and
   can span multiple lines.
*/

#include <stdio.h>

int main() {

    int x = 5;
    x = 10;

    float y;
    y = 20.5;

    printf("x is %i\n", x);
    printf("y is %f\n", y);

    printf("Enter a new value for x: ");
    scanf("%i", &x);

    printf("x is now: %i\n", x);

    return 0;
}
```

- Anything that comes after `//` is ignored by the compiler (gcc)
- Anything between the following symbols will be ignored by the compiler and can span multiple lines.
 - `/**/`

main()

```
// This is a line comment. Anything that comes after // is ignored by the compiler (gcc)

/*
   This is a block comment.
   Anything between the symbols will be ignored by the compiler and
   can span multiple lines.
*/

#include <stdio.h>

int main() {
    int x = 5;
    x = 10;

    float y;
    y = 20.5;

    printf("x is %i\n", x);
    printf("y is %f\n", y);

    printf("Enter a new value for x: ");
    scanf("%i", &x);

    printf("x is now: %i\n", x);

    return 0;
}
```

- Execution of a C program starts in the main() function.
- The book uses int main(void) but the void is optional (and it's redundant in this case).

Variables: Declaration and Initialization

```
// This is a line comment. Anything that comes after // is ignored by the compiler (gcc)

/*
   This is a block comment.
   Anything between the symbols will be ignored by the compiler and
   can span multiple lines.
*/

#include <stdio.h>

int main() {
    int x = 5;
    x = 10;

    float y;
    y = 20.5;

    printf("x is %i\n", x);
    printf("y is %f\n", y);

    printf("Enter a new value for x: ");
    scanf("%i", &x);

    printf("x is now: %i\n", x);

    return 0;
}
```

- Variables require a type and the types cannot change.
- Variables store values in memory (RAM).

Variables: Declaration and Initialization

```
// This is a line comment. Anything that comes after // is ignored by the compiler (gcc)

/*
   This is a block comment.
   Anything between the symbols will be ignored by the compiler and
   can span multiple lines.
*/

#include <stdio.h>

int main() {

    int x = 5;
    x = 10;

    float y;
    y = 20.5;

    printf("x is %i\n", x);
    printf("y is %f\n", y);

    printf("Enter a new value for x: ");
    scanf("%i", &x);

    printf("x is now: %i\n", x);

    return 0;
}
```

- A variable's value can change.

Variables: Declaration and Initialization

```
// This is a line comment. Anything that comes after // is ignored by the compiler (gcc)

/*
   This is a block comment.
   Anything between the symbols will be ignored by the compiler and
   can span multiple lines.
*/

#include <stdio.h>

int main() {

    int x = 5;
    x = 10;

    float y;
    y = 20.5;

    printf("x is %i\n", x);
    printf("y is %f\n", y);

    printf("Enter a new value for x: ");
    scanf("%i", &x);

    printf("x is now: %i\n", x);

    return 0;
}
```

- Declares a variable y with type float. The value of y is uninitialized
- this means the value is unknown (undefined) and can be any valid float variable.

Display text to the terminal window

```
// This is a line comment. Anything that comes after // is ignored by the compiler (gcc)

/*
   This is a block comment.
   Anything between the symbols will be ignored by the compiler and
   can span multiple lines.
*/

#include <stdio.h>

int main() {

    int x = 5;
    x = 10;

    float y;
    y = 20.5;

    printf("x is %i\n", x);
    printf("y is %f\n", y);

    printf("Enter a new value for x: ");
    scanf("%i", &x);

    printf("x is now: %i\n", x);

    return 0;
}
```

- We can print strings and variable.
- The %i conversion specification is used to specify that we want to substitute that for an integer value and %f for float

Input from user

```
// This is a line comment. Anything that comes after // is ignored by the compiler (gcc)

/*
   This is a block comment.
   Anything between the symbols will be ignored by the compiler and
   can span multiple lines.
*/

#include <stdio.h>

int main() {

    int x = 5;
    x = 10;

    float y;
    y = 20.5;

    printf("x is %i\n", x);
    printf("y is %f\n", y);

    printf("Enter a new value for x: ");
    scanf("%i", &x);

    printf("x is now: %i\n", x);

    return 0;
}
```

- scanf() reads input from the user.
- We specifically want an integer (because of the %i).
- To store the value in x we need to use &x.
- Do not use "\n" for scanf().

return

```
// This is a line comment. Anything that comes after // is ignored by the compiler (gcc)

/*
   This is a block comment.
   Anything between the symbols will be ignored by the compiler and
   can span multiple lines.
*/

#include <stdio.h>

int main() {

    int x = 5;
    x = 10;

    float y;
    y = 20.5;

    printf("x is %i\n", x);
    printf("y is %f\n", y);

    printf("Enter a new value for x: ");
    scanf("%i", &x);

    printf("x is now: %i\n", x);

    return 0;
}
```

- By convention returning 0 means everything ran okay. Non-zero values are used for errors.

Compile and Run

```
// This is a line comment. Anything that comes after // is ignored by the compiler (gcc)

/*
   This is a block comment.
   Anything between the symbols will be ignored by the compiler and
   can span multiple lines.
*/

#include <stdio.h>

int main() {

    int x = 5;
    x = 10;

    float y;
    y = 20.5;

    printf("x is %i\n", x);
    printf("y is %f\n", y);

    printf("Enter a new value for x: ");
    scanf("%i", &x);

    printf("x is now: %i\n", x);

    return 0;
}
```

```
x is 10
y is 20.500000
```

Compile and Run

```
// This is a line comment. Anything that comes after // is ignored by the compiler (gcc)

/*
   This is a block comment.
   Anything between the symbols will be ignored by the compiler and
   can span multiple lines.
*/

#include <stdio.h>

int main() {

    int x = 5;
    x = 10;

    float y;
    y = 20.5;

    printf("x is %i\n", x);
    printf("y is %f\n", y);

    printf("Enter a new value for x: ");
    scanf("%i", &x);

    printf("x is now: %i\n", x);

    return 0;
}
```

Enter a new value for x: 10.5
x is now: 10