

- A First C Program
 - Real World Application: Computing Distances
 - Identifiers
-
- Redirecting Input and Output
 - Files

```
#include <stdio.h>
main()
{
    int distance, rate, time;
    rate = 14;
    printf( "Enter next time: " )
    scanf( "%f", time);
    while ( time > 0 ) {
        distance = rate * time;
        printf( "Time = %d hours\n", time);
        printf( "Distance = %d kilometers\n", distance);
        printf( "Enter next time: " );
        scanf( "%d", &time );
    }
    printf( "*** End of Program ***\n" );
}
/* file name distance.c */
```

Redirecting Input and Output

standard input ---> keyboard
 standard output ---> video } Normal

Input redirection ---> the process of changing the standard input.

Output redirection ---> the process of changing the standard output.

Note: Every system has some method of redefining the standard input and the standard output.

In MS-DOS or UNIX,

System prompt	Executable file	Redirect input	Input file
↓	↓	↙	↘
	<u>% distance</u>	<	<u>data.in</u>
	<u>% distance</u>	>	<u>data.out</u>
		↑	↙
		Redirect output	output file

Files

Old

scanf () --> standard input --> Keyboard

printf () --> standard output --> video

or

scanf () --> redirecting input --> input file

printf () --> redirecting output --> output file

```
#include <stdio.h>
main()
{
    float distance, rate, time;
    FILE *fin, *fout;
    fin = fopen( "datain.txt", "r");
    fout = fopen ( "data.out", "w");
    rate = 14.0;
    while ( fscanf( fin, "%f", &time) != EOF ) {
        distance = rate * time;
        fprintf( fout, "Time = %f hours\n", time);
        fprintf( fout, "Distance = %f kilometers\n\n", distance);

    }
    printf( "*** End of Program ***\n" );
    fclose( fin);
    fclose( fout);
}
/* file name distance.c */
```

New

```
FILE *fin, *fout; /*define fin and fout as file pointers*/  
fin = fopen( "data.in", "r"); /*open data.in for reading*/  
fout = fopen( "data.out", "w"); /*open data.out for writing*/
```

Each file must be handled with a file pointer

```
fopen("filename","r")  
"w")  
"a")
```

```
fscanf  
fprintf
```

Things to remember

- Use fscanf, not scanf
- use fprintf, not printf
- fclose(fin)
- fclose(fout)
- to be discussed further:

– FILE *fin,*fout;

Structure

file pointers

Real World Application: Classify Solutions as Acidic or Nonacidic

Problem

Molar concentration

$$pH = -\log_{10} mc$$



- pH < 7, acidic
- pH = 7, neutral
- pH > 7, nonacidic or alkaline

Write a program that reads mc from input file, writes the corresponding pH, and classifies the solution as acidic or nonacidic.

```
#include <stdio.h>
#include <math.h>
main ( )
{
    float mc, ph;
    FILE *fin, *fout;
    fin = fopen ( "ph-infile.txt", "r");
    fout = fopen ( "ph-outfile.txt", "w");
    while ( fscanf ( fin, " %f", &mc ) != EOF )
    {
        ph = -log10( mc );
        fprintf ( fout, "\nMolar concentration = %e\n", mc);
        fprintf ( fout, "pH = %f\n", ph );
        if ( ph < 7.0 )
            fprintf ( fout, "Acidic\n");
    }
}
```

```
    else  
        fprintf (fout, "Nonacidic\n");  
    }  
}
```