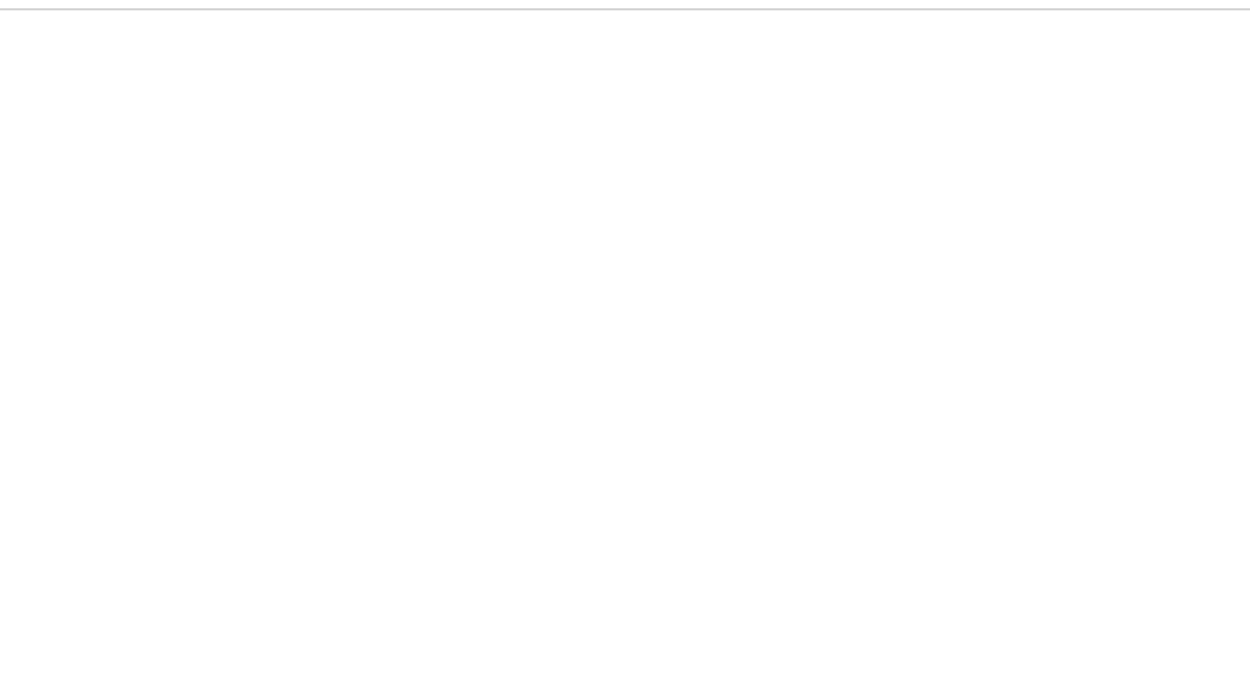
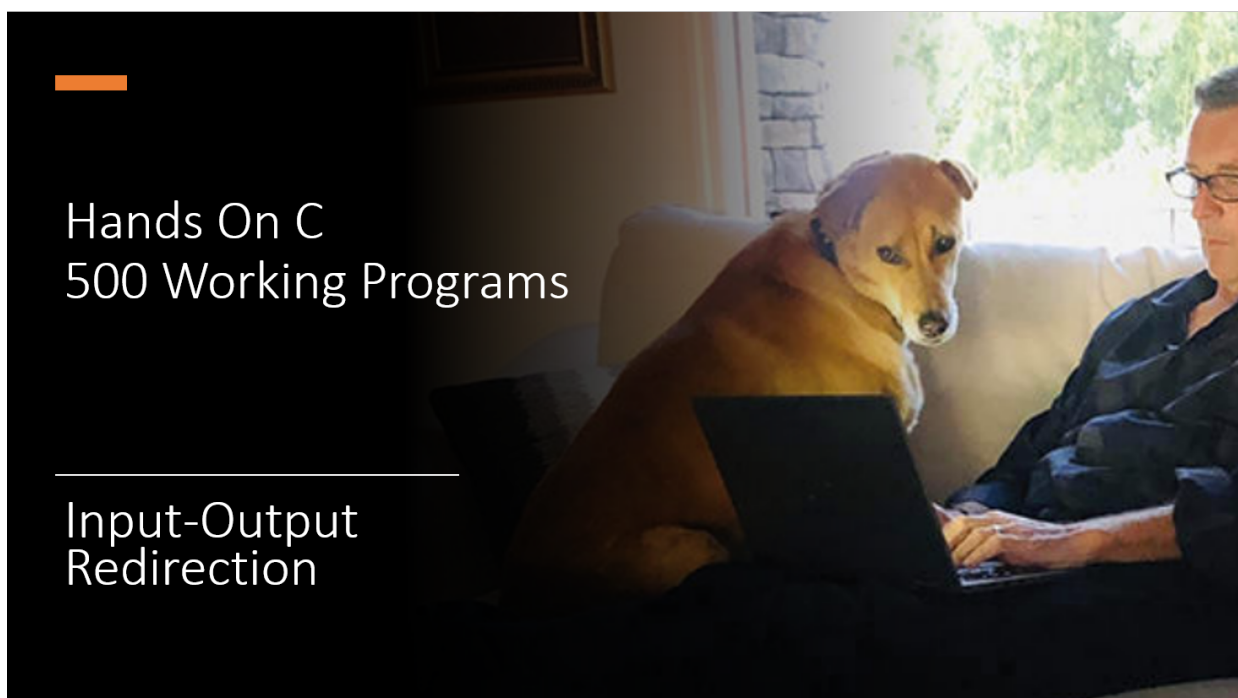




Understanding I/O Redirection

```
C:\> DIR > Files.txt <Enter>

C:\> MORE < Files.txt <Enter>

C:\> DIR >> Append.txt <Enter>

C:\> DIR | MORE <Enter>
```

C Programs Use Common File Handles

Handles	Default
stdin	Keyboard
stdout	Console screen
stderr	Console screen
stdprn	Printer

In [1]: `#include <stdio.h>`

```
int main(void)
{
    fputs("Hello, World!", stdout);
}
```

Hello, World!

In [2]: `#include <stdio.h>`

```
int main(void)
{
    fputs("Standard output", stdout);
    fputs("Standard error", stderr);
}
```

Standard output

Standard error

```
C:\> Program > File.txt <Enter>
Standard Error
```

```
C:\>
```

Displaying Redirected Input in Uppercase

```
In [3]: #include <stdio.h>
#include <stdlib.h>

int main(void)
{
    FILE *fp = fopen("Alphabet.txt", "r");

    char letter;

    while ((letter = fgetc(fp)) != EOF)
        fputc(letter, stdout);

    fclose(fp);
}
```

ABCDEFGHIJKLMNOPQRSTUVWXYZ

```
In [4]: #include <stdio.h>
#include <ctype.h>

int main(void)
{
    char letter;

    while ((letter = fgetc(stdin)) != EOF)
        fputc(toupper(letter), stdout);
}
```

```
In [5]: #include <stdio.h>
#include <ctype.h>

int main(void)
{
    char letter;

    while ((letter = getchar()) != EOF)
        putchar(toupper(letter));
}
```

Preceding Redirected Input with a Line Number

```
In [6]: #include <stdio.h>

int main(void)
{
    char line[256];
    int line_number = 0;

    while (fgets(line, sizeof(line), stdin))
        printf("%d %s", ++line_number, line);
}
```

Displaying a Count of Redirected Input

```
In [7]: #include <stdio.h>

int main(void)
{
    char line[256];
    int line_number = 0;

    while (fgets(line, sizeof(line), stdin))
        line_number++;

    printf("Redirected lines %d\n", line_number);
}
```

Redirected lines 0

Splitting Redirected Input to the Screen and Printer

In [8]: `#include <stdio.h>`

```
int main(void)
{
    char line[256];

    while (fgets(line, sizeof(line), stdin))
    {
        fputs(line, stdout);
        fputs(line, stdprn);
    }
}
```

/tmp/tmp5wmfvk_3.c: In function 'main':

/tmp/tmp5wmfvk_3.c:10:18: error: 'stdprn' undeclared (first use in this function); did you mean 'stderr'?

```
    fputs(line, stdprn);
```

```
                ^~~~~~
```

```
                stderr
```

/tmp/tmp5wmfvk_3.c:10:18: note: each undeclared identifier is reported only once for each function it appears in

[C kernel] GCC exited with code 1, the executable will not be executed

Displaying the First n Lines of Redirected Input

```
In [9]: #include <stdio.h>
#include <stdlib.h>

void main(int argc, char *argv[])
{
    char line[255];
    int i, j;

    if (argc < 2)
        j = 10;
    else
        j = atoi(argv[1]);

    for (i=0; i < j; i++)
    {
        if (fgets(line, sizeof(line), stdin) == NULL)
            break;
        fputs(line, stdout);
    }
}
```

What You will Learn Next

C arrays let you store multiple values of the same type. Common operations C programmers must perform are to search an array for a specific value and to sort an array in ascending or descending order.

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

```
#define ARRAY_SIZE 5

int main(void)
{
    char *values[ARRAY_SIZE] = { "AAA", "CCC", "DDD", "BBB", "EEE" };

    for (int i = 0; i < ARRAY_SIZE; ++i)
        if (strcmp(values[i], "BBB") == 0)
            printf("Found BBB at index %d\n", i);
}
```