File IO Part II

fgets, fputs, fgetc, fputc

Get and Put a Line

```
#include <stdio.h>
char *fgets(char *s, int n, FILE * fp);
int fputs(char *s, FILE * fp);
```

- These two functions read or write a string from or to a file.
- fgets reads an entire line into s, up to n-1 characters in length (pass the size of the character array s in as n to be safe!)
- fgets returns the pointer s on success, or NULL if an error or end-of-file is reached.
- fputs returns the number of characters written if successful; otherwise, return EOF.

Reading lines from a file: fgets()

- Takes three parameters
  - a character array str, maximum number of characters to read size, and a file pointer fp
- Reads from the file fp into the array str until any one of these happens
  - No. of characters read = size - 1
  - \n is read (the char \n is added to str)
  - EOF is reached or an error occurs
- '\0' added at end of str if no error
- Returns NULL on error or EOF, otherwise returns pointer to str

Reading lines from a file:
fgets()

```
FILE *fptr;
char line[1000];
/* Open file and check it is open */
while (fgets(line,1000,fptr) != NULL)
{
   printf ("Read line %s\n",line);
}
```

Writing lines to a file: fputs()

- Takes two parameters
  - A string str (null terminated) and a file pointer fp
- Writes the string pointed to by str into the file
- Returns non-negative integer on success, EOF on error

A Sample program to show the use of fputs, fgets

```c
#include <stdio.h>

int main()
{
    FILE *fptr = fopen("data.txt","w");
    fputs("CProgramming\n",fptr);
    fputs("JavaProgramming",fptr);
    fclose(fptr);

    fptr = fopen("data.txt","r");
    char ch[20];
    while(fgets(ch,20,fptr)!=NULL)
    {
        printf("%s",ch);
    }

    fclose(fptr);

    return 0;
}
```

Program Output

Get and Put a Character

```
#include <stdio.h>
                    int fgetc(FILE * fp);
                    int fputc(int c, FILE * fp);
```

- These two functions read or write a single byte from or to a file.
- fgetc returns the character that was read, converted to an integer.
- fputc returns the same value of parameter c if it succeeds, otherwise, return EOF.

Reading/Writing a character: fgetc(), fputc()

- Equivalent of getchar(), putchar() for reading/writing char from/to keyboard
- Exactly same, except that the first parameter is a file pointer
- Equivalent to reading/writing a byte (the char)

```
char fgetc(FILE *fp);
int fputc(char c, FILE *fp);
```

- Example:

```
char c;
c = fgetc(fp1); fputc(c, fp2);
```

A Sample program to show the use of putc,getc

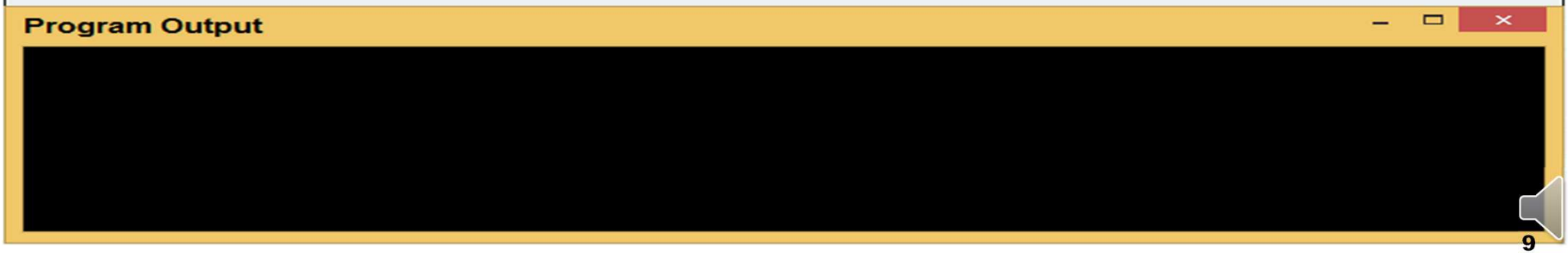
```c
#include <stdio.h>
#include <string.h>

int main()
{
    FILE *fptr = fopen("data.txt","w");
    char str[] = "C Programming";
    int i=0;
    while(i<strlen(str))
    {
        putc(str[i],fptr);
        i++;
    }
    fclose(fptr);

    fptr = fopen("data.txt","r");
    char c;
    while((c=getc(fptr))!=EOF)
    {
        printf("%c",c);
    }
    fclose(fptr);

    return 0;
}
```

Program Output



A Sample program to show the use of fgetc

The fgetc function Returns the character from the specified FILE.

```c
#include <stdio.h>

int main()
{
      FILE *ptr=fopen("ha.dat","w");
      fprintf(ptr,"WELCOME");
      fclose(ptr);

      ptr=fopen("ha.dat","r");

      char a;
      a=fgetc(ptr);
      printf("%c\n",a);

      a=fgetc(ptr);
      printf("%c\n",a);

      a=fgetc(ptr);
      printf("%c\n",a);

      fclose(ptr);
      return 0;
}
```

program output

```
W
E
L
```