

Experiment g-1 (File Handling)

1) ~~#include~~
→ #include <stdio.h>

```
int main() {  
    file *fp;  
    char text[100];
```

```
    fp = fopen ("myfile.txt", "w");  
    if (fp == NULL) {  
        printf ("Error Opening file: /n");  
        return 1;  
    }
```

```
    printf ("Enter text to write into the file: /n");  
    fgets (text, size of (text), stdin);
```

```
    fputs (text, fp);  
    fclose (fp);
```

```
    printf ("Text written to 'myfile.txt' successfully. /n");  
    return 0;
```

Remarks:

Teacher's Signature _____

The screenshot shows the Visual Studio Code interface with a dark theme. The top bar includes tabs for 'Welcome' (active), 'Untitled-1' (C file), 'Settings', and other icons. The main editor area contains the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     FILE *fp;
5     char text[200];
6
7     fp = fopen("sample.txt", "w");
8
9     if (fp == NULL) {
10         printf("Error opening file!\n");
11         return 1;
12     }
13
14     printf("Enter text to write into file: ");
15     fgets(text, sizeof(text), stdin);
16
17     fputs(text, fp);
18     fclose(fp);
19
20     printf("File created and text written successfully.\n");
21
22     return 0;
23 }
```

The bottom section shows the 'TERMINAL' tab active, displaying terminal output:

```
cd "/Users/rajatsingh/main.c/" && gcc tempCodeRunnerFile.c -o tempCodeRunne
rFile && "/Users/rajatsingh/main.c/">tempCodeRunnerFile
● rajatsingh@Rajats-MacBook-Air main.c % cd "/Users/rajatsingh/main.c/" && gc
c tempCodeRunnerFile.c -o tempCodeRunnerFile && "/Users/rajatsingh/main.c/">
tempCodeRunnerFile
Enter text to write into file: hello i am Rajat
File created and text written successfully.
○ rajatsingh@Rajats-MacBook-Air main.c %
```

A dropdown menu in the terminal area shows options: zsh (selected) and Code.