

PRACTICAL NO:-4

Aim:- To explain compile time errors and runtime errors.

1. Compile Time Errors

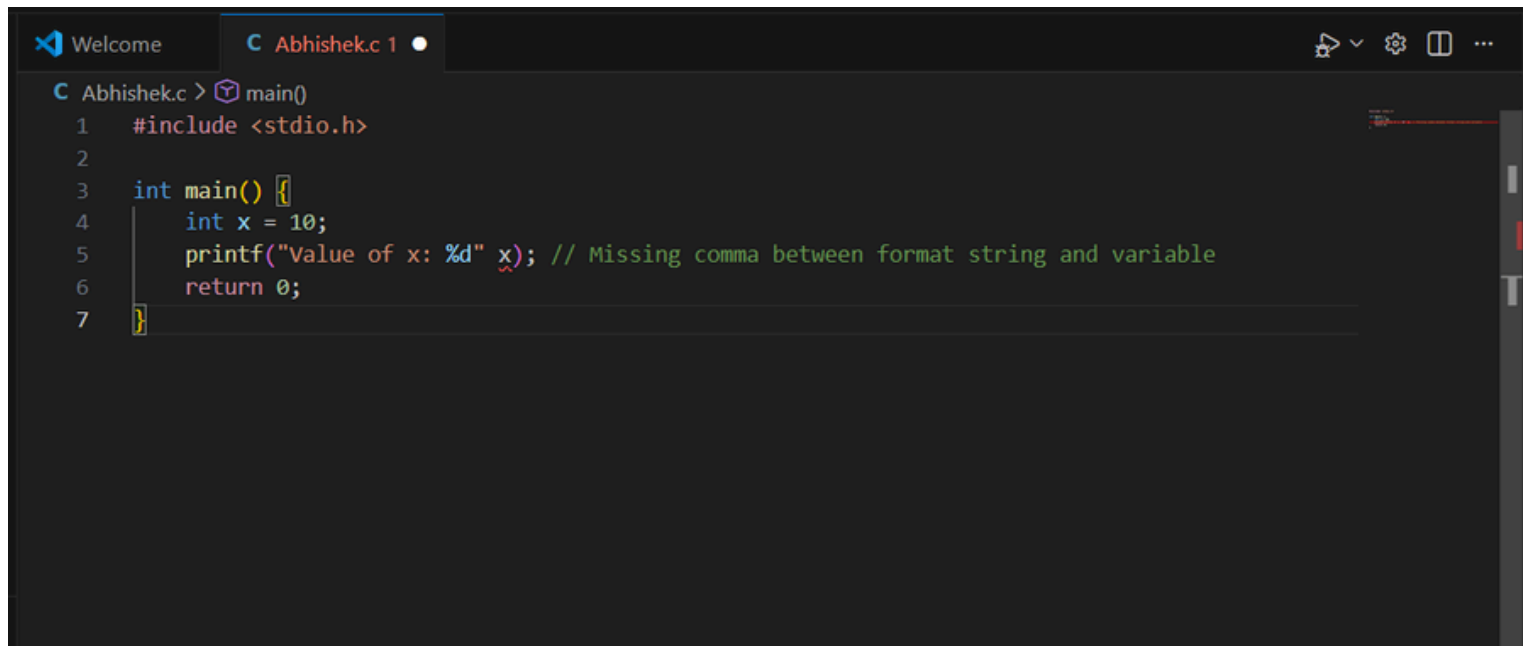
- **These errors occur when the program is being compiled.**
- **The compiler detects these errors before the program is executed.**
- **Common examples include:**
- **Syntax errors**
- **Missing semicolons**
- **Undeclared variables**
- **Type mismatches**
- **Missing brackets, etc.**
- **The program does not run until all compile time errors are corrected.**

2. Run Time Errors

- **These errors occur while the program is running (after successful compilation).**
- **They are usually caused by:**
- **Invalid input**
- **Division by zero**
- **File handling issues**
- **Memory-related issues**
- **The program may compile successfully but can crash during execution due to runtime errors.**

Example of Compile Time Error

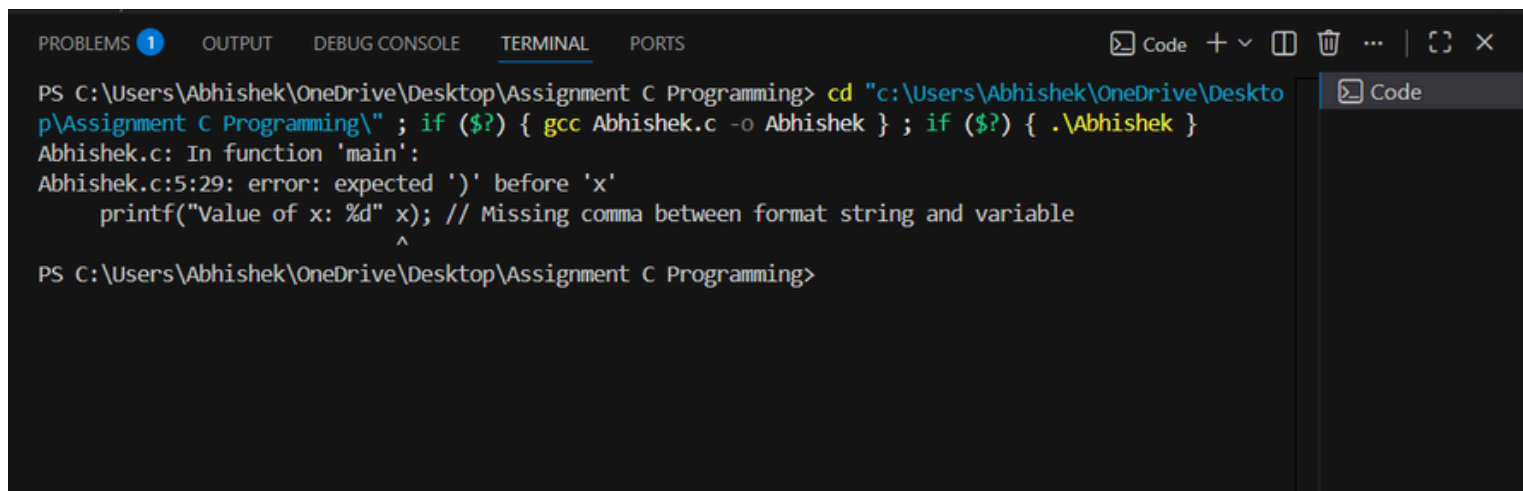
- **CODE :-**



The screenshot shows a code editor with a file named 'Abhishek.c'. The code is as follows:

```
1  #include <stdio.h>
2
3  int main() {
4      int x = 10;
5      printf("Value of x: %d" x); // Missing comma between format string and variable
6      return 0;
7  }
```

- **OUTPUT :-**



The screenshot shows a terminal window with the following output:

```
PS C:\Users\Abhishek\OneDrive\Desktop\Assignment C Programming> cd "c:\Users\Abhishek\OneDrive\Desktop\Assignment C Programming\" ; if ($?) { gcc Abhishek.c -o Abhishek } ; if ($?) { .\Abhishek }
Abhishek.c: In function 'main':
Abhishek.c:5:29: error: expected ')' before 'x'
    printf("Value of x: %d" x); // Missing comma between format string and variable
                          ^
PS C:\Users\Abhishek\OneDrive\Desktop\Assignment C Programming>
```

- **Explanation: -** The error occurs because the printf statement is written incorrectly. Since the syntax is wrong, the compiler stops and refuses to generate an executable program.

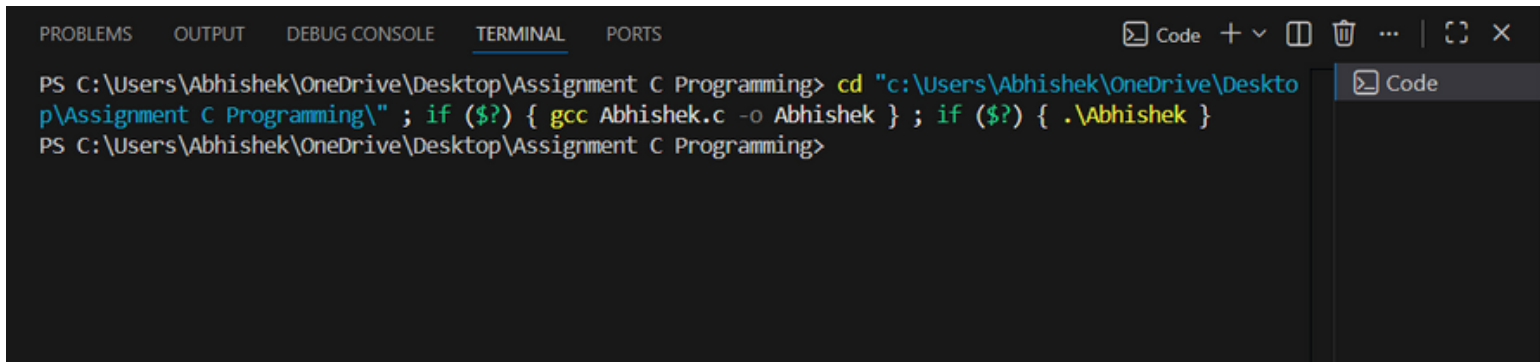
Example of Run Time Error

- **CODE :-**



```
Abhishek.c > main()
1  #include <stdio.h>
2
3  int main() {
4      int numerator = 10;
5      int denominator = 0;
6      int result = numerator / denominator; // Division by zero
7      printf("Result: %d\n", result);
8      return 0;
9  }
```

- **OUTPUT :-**



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Abhishek\OneDrive\Desktop\Assignment C Programming> cd "c:\Users\Abhishek\OneDrive\Desktop\Assignment C Programming\" ; if ($?) { gcc Abhishek.c -o Abhishek } ; if ($?) { .\Abhishek }
PS C:\Users\Abhishek\OneDrive\Desktop\Assignment C Programming>
```

- **Explanation:-** The code is syntactically correct, so it compiles successfully.

However, dividing a number by zero causes a runtime error, leading to abnormal termination.