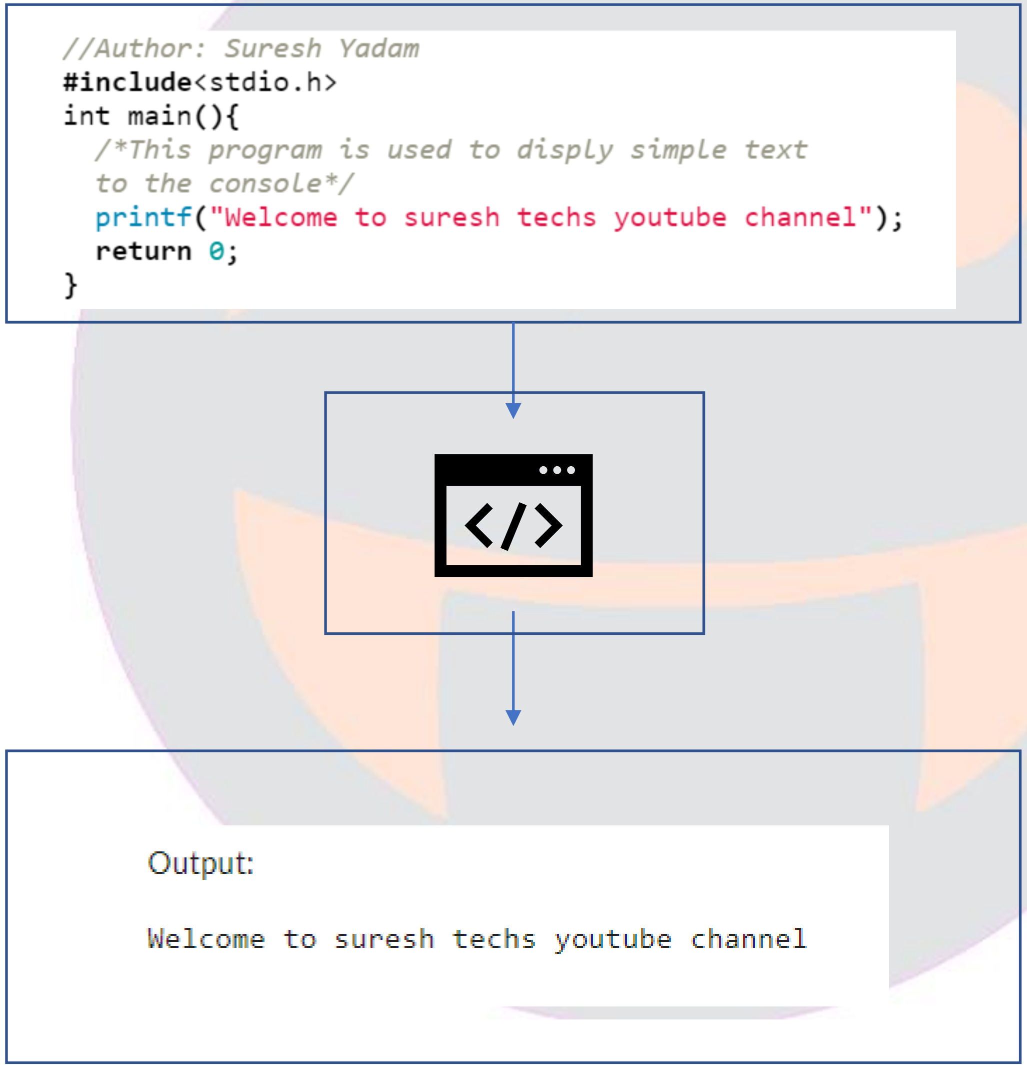
Executing a C program

CHAPTER 7

SURESH TECHS

C PROGRAMMING COURSE

Executing a c program



```c
//Author: Suresh Yadam
#include<stdio.h>
int main(){
  /*This program is used to disply simple text
  to the console*/
  printf("Welcome to suresh techs youtube channel");
  return 0;
}
```

Output:

```
Welcome to suresh techs youtube channel
```

Input – **Source Code File**/Program(welcome.c)

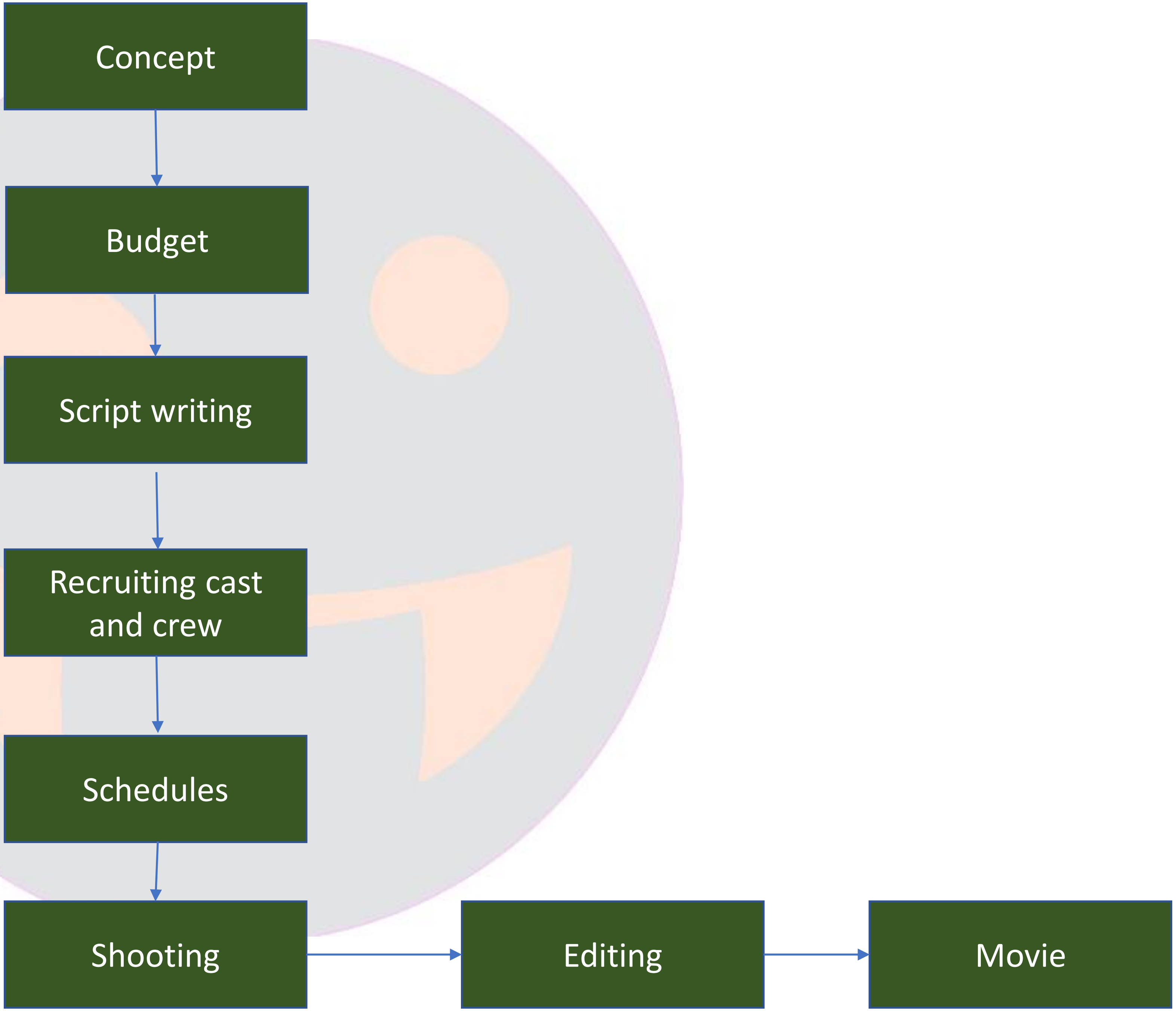
What is happening here 🙄 ? Who is this? Who is **executing** our program?

Output of the program, **Executable file**

Movie(RRR)

You may not remember all these terms, but you the actual process of how a movie is made from beginning to end

Similarly, you may not remember all The steps involved in executing a c programming for the first time.
Don't worry



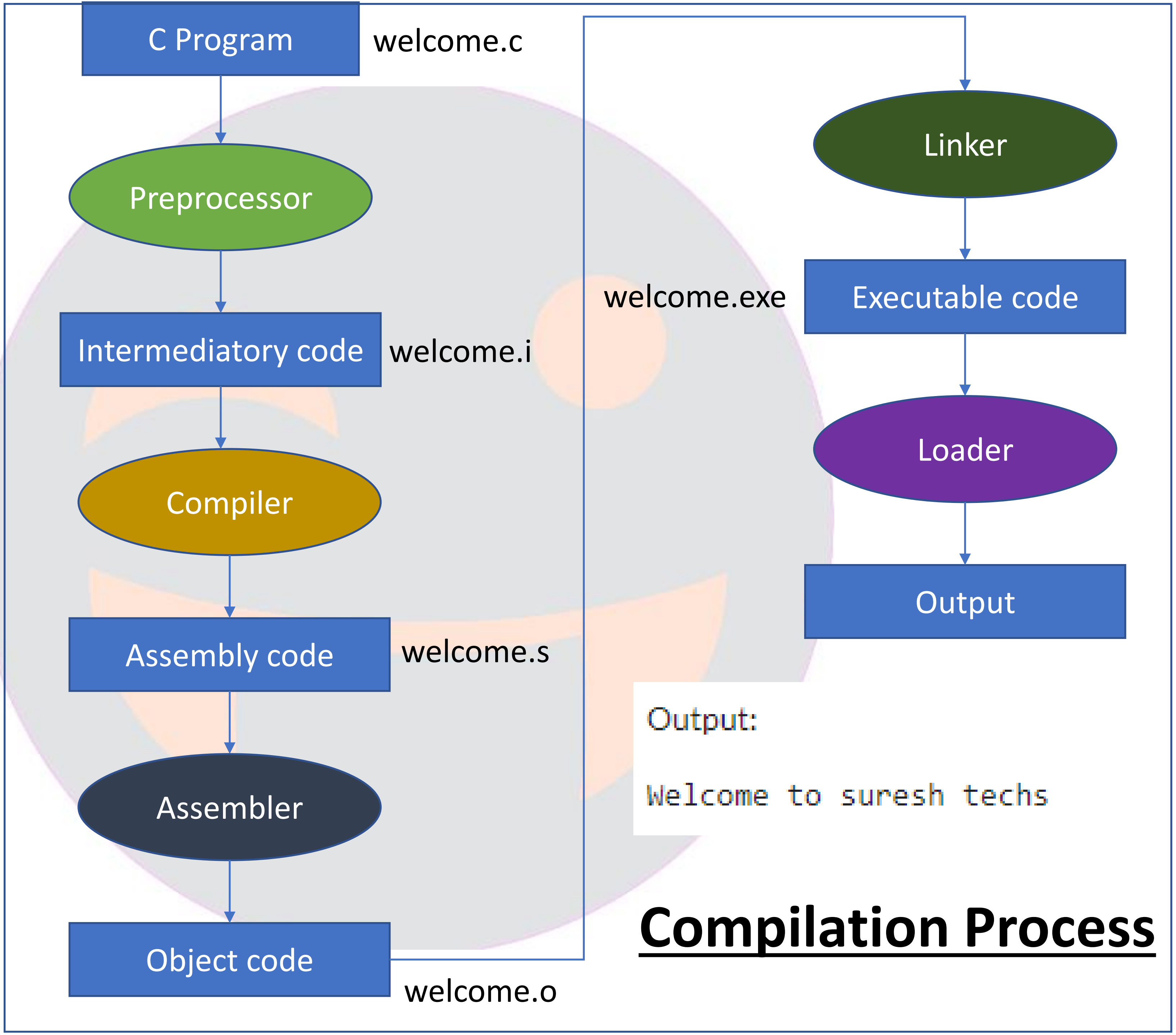
Executing C Program

Who will do this compilation process?

COMPILER

Do I need a compiler to execute my c program?

YES



C Program welcome.c

Preprocessor

Intermediatory code welcome.i

Compiler

Assembly code welcome.s

Assembler

Object code welcome.o

Linker

welcome.exe Executable code

Loader

Output

Output:

```
Welcome to suresh techs
```

<u>Compilation Process</u>

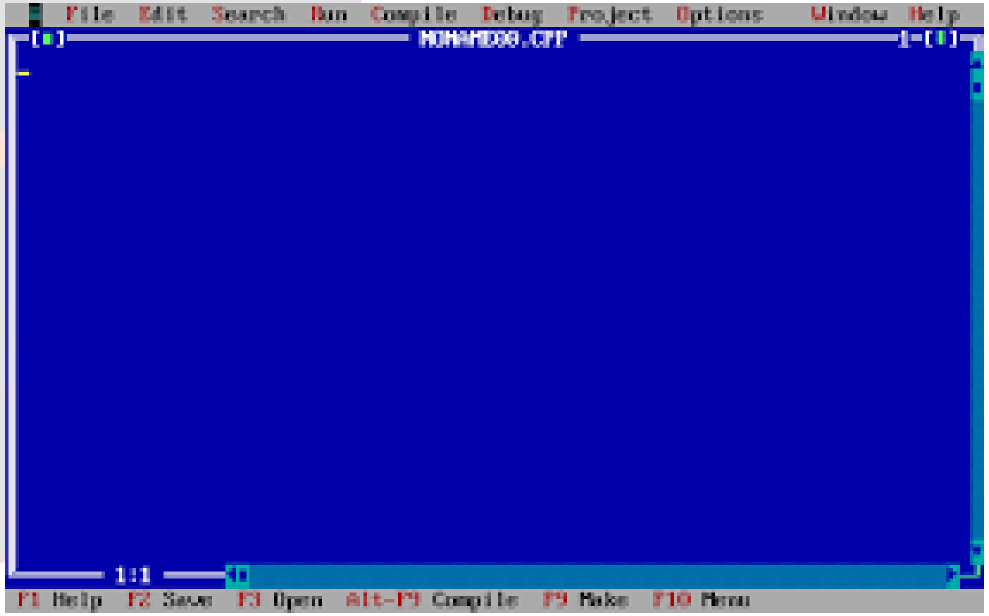
Compilation

- The compilation is a process of converting the **source code** into **object code**. It is done with the help of the compiler

Different C Compilers

Compiler	Release	Developer
pccm	1973	Bell Labs
BSD C	1979	Zolman
Aztec C	1980	Manx Software Systems
ACK	1980	Tanenbaum, Jacobs
Lattice C	1982	Steve Krueger
MPW	1986	Apple
GCC	1987	GNU Project
Turbo C	1987	Turbo
Megamax C	1988	Megamax, Inc
Acorn C	1988	Acorn
LabWindows	1989	National Instruments
QuickC	1990	Microsoft
Oracle C	1991	Oracle
MinGW	1993	Peters

GNU Compiler Collection

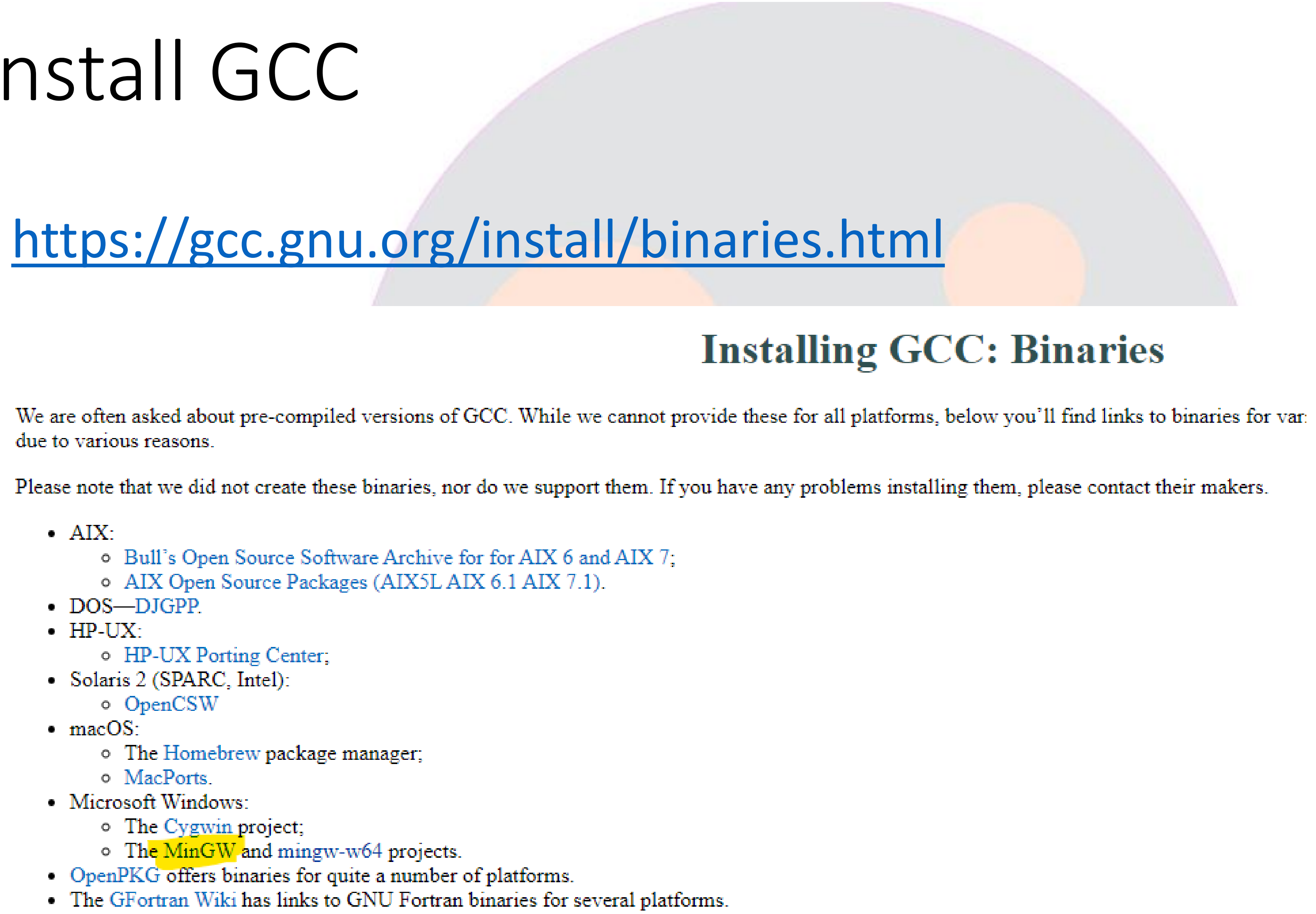


Installing GCC

- Note: Just watch, don’t install it. We will use an IDE(Code blocks) later to run our programs
- I will just install GCC and explain how the compilation process looks like

Install GCC

- https://gcc.gnu.org/install/binaries.html



Installing GCC: Binaries

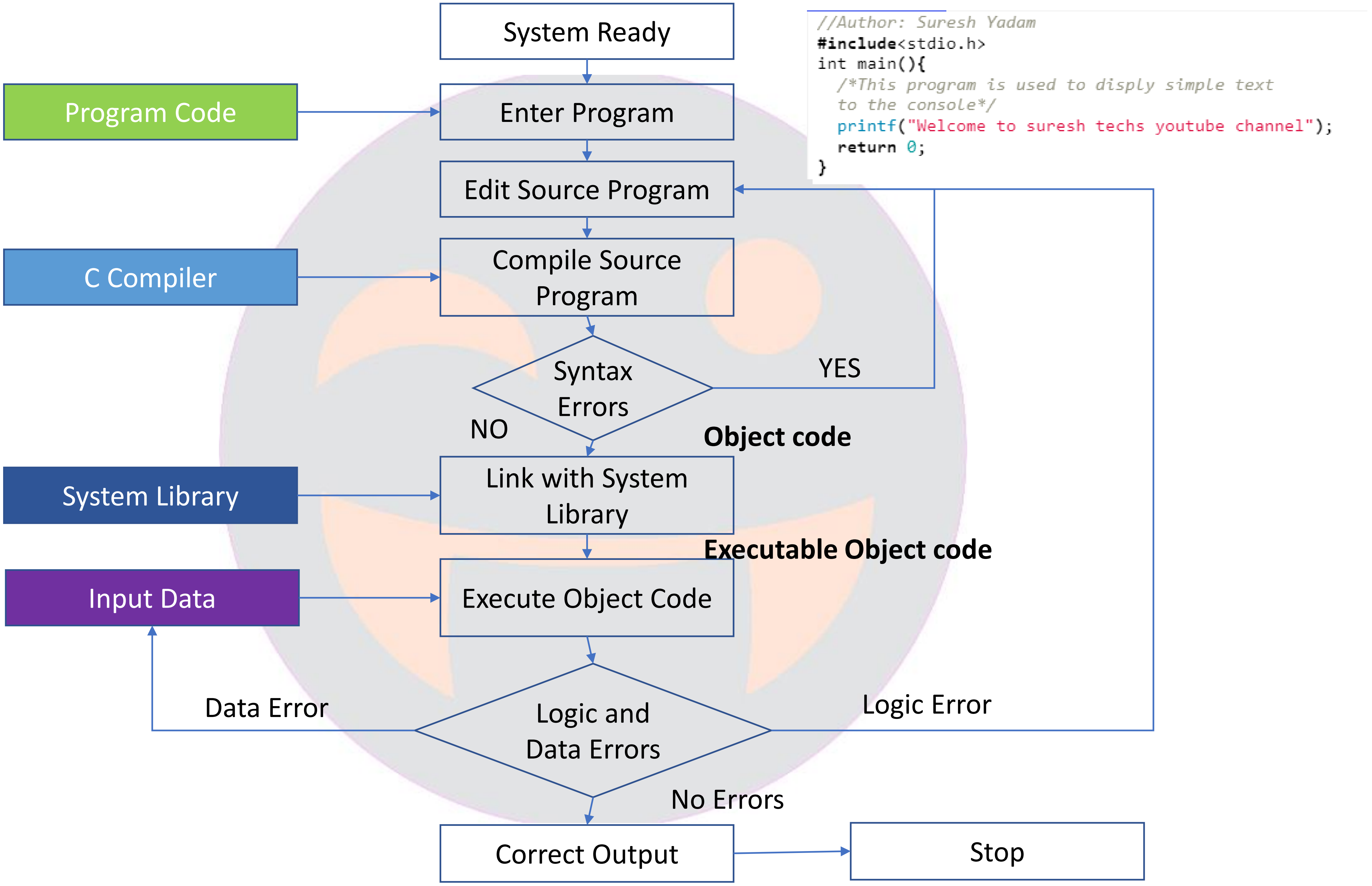
We are often asked about pre-compiled versions of GCC. While we cannot provide these for all platforms, below you'll find links to binaries for var: due to various reasons.

Please note that we did not create these binaries, nor do we support them. If you have any problems installing them, please contact their makers.

- AIX:
  - Bull's Open Source Software Archive for for AIX 6 and AIX 7;
  - AIX Open Source Packages (AIX5L AIX 6.1 AIX 7.1).
- DOS—DJGPP.
- HP-UX:
  - HP-UX Porting Center;
- Solaris 2 (SPARC, Intel):
  - OpenCSW
- macOS:
  - The Homebrew package manager;
  - MacPorts.
- Microsoft Windows:
  - The Cygwin project;
  - The MinGW and mingw-w64 projects.
- OpenPKG offers binaries for quite a number of platforms.
- The GFortran Wiki has links to GNU Fortran binaries for several platforms.

Executing C program

- Creating the program
- Compiling the program
- Linking the program with functions that are needed from the C library(header files)
- Executing the program
- Let’s check the flow diagram



```
//Author: Suresh Yadam
#include<stdio.h>
int main(){
  /*This program is used to disply simple text
  to the console*/
  printf("Welcome to suresh techs youtube channel");
  return 0;
}
```

What next?

- Install an IDE (Code::Blocks) and write our first program