UPES

UNIVERSITY WITH A PURPOSE

UPES

UNIVERSITY WITH A PURPOSE

Compiler Construction

Presented by:

Krishan Rupani	Neha Singh	Payal Singla
500067535	500069028	500069630
R171218057	R171218067	R171218118

Under the guidance of:

Dr. Hitesh Kumar Sharma

Assistance Professor

Department of Cybernetics,

School of Computer Science

UPES

UNIVERSITY WITH A PURPOSE

Introduction:

- Compiler is not a unknown word to all of us, compiler is an integral part of our programming journey.
- It converts our program written in any language to machine understandable code.
- So a programmer writes the code in the compiler and compiler produces output according to the code or it produces error if the code is incorrect in anyway.
- We are making a online web based compiler which will contain most of the language, in future and will reduce the trouble of installing many different compilers on your PC.
- Inspiration to this project came from my own suffering, I have a PC with good computational power but because I do a lot stuff managing so many different compilers for every language made my life really tough.
- Another problem that I have noticed is many edtech startup wishes to have a in build compiler on their web application, we are trying to solve that problem.

UPES
UNIVERSITY WITH A PURPOSE

Objective:

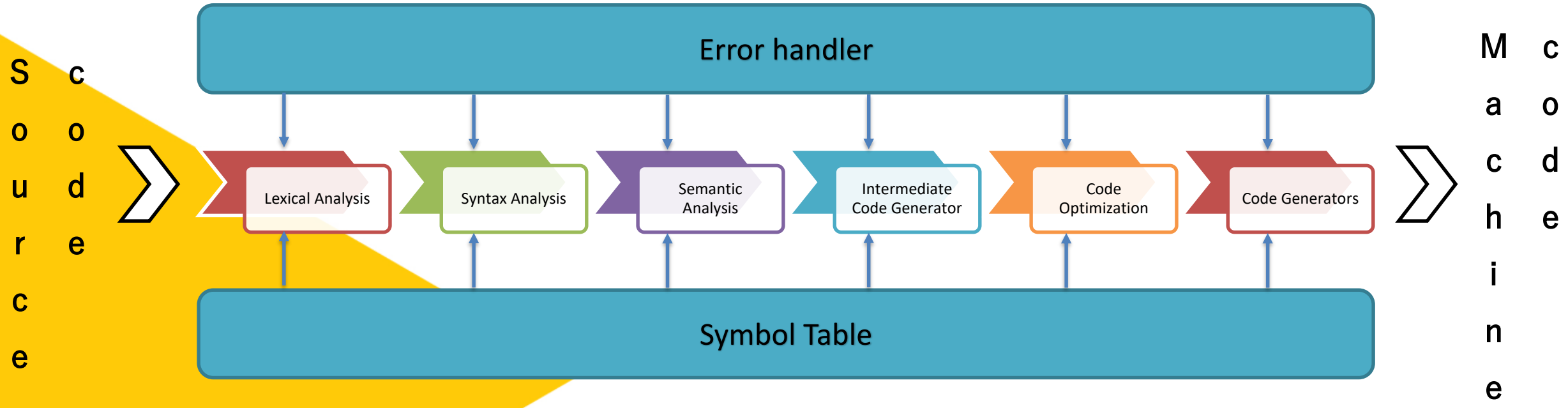
Our aim for this minor project 1 is:

To create a web based compiler for subset of C language.

UPES
UNIVERSITY WITH A PURPOSE

How Compiler Works:

In order to convert the source code in machine understandable code, the source code has to pass 6 phases.

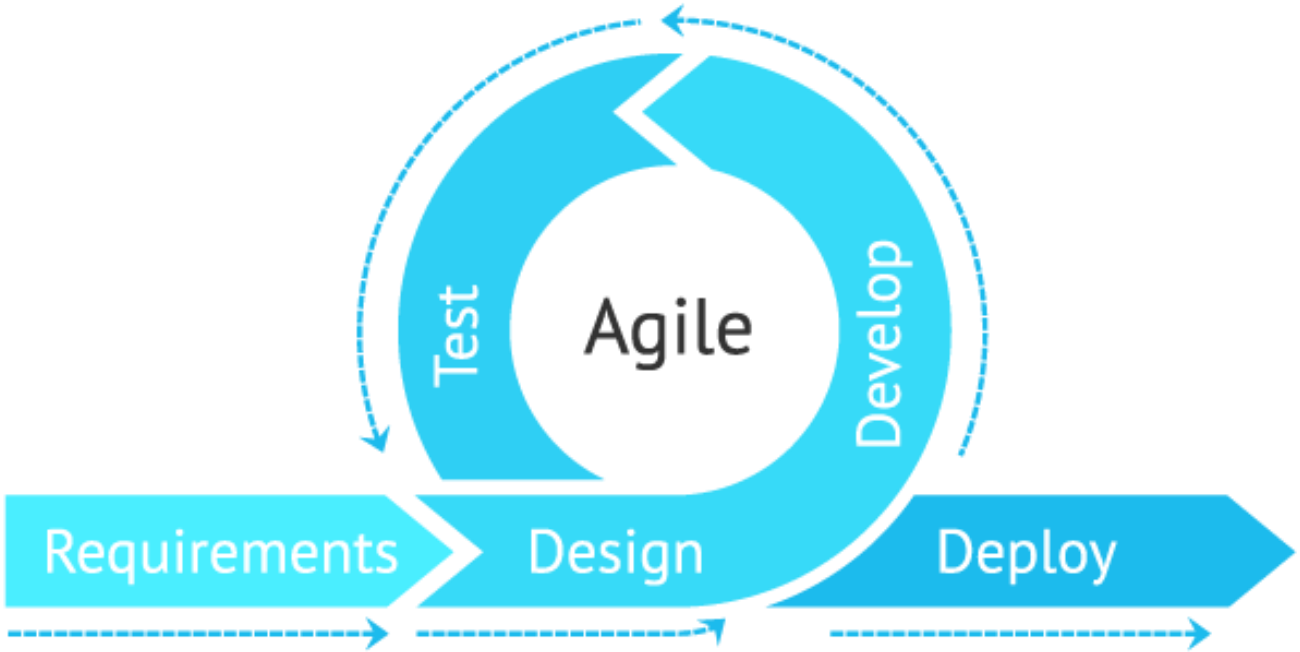


UPES
UNIVERSITY WITH A PURPOSE

Methodology:

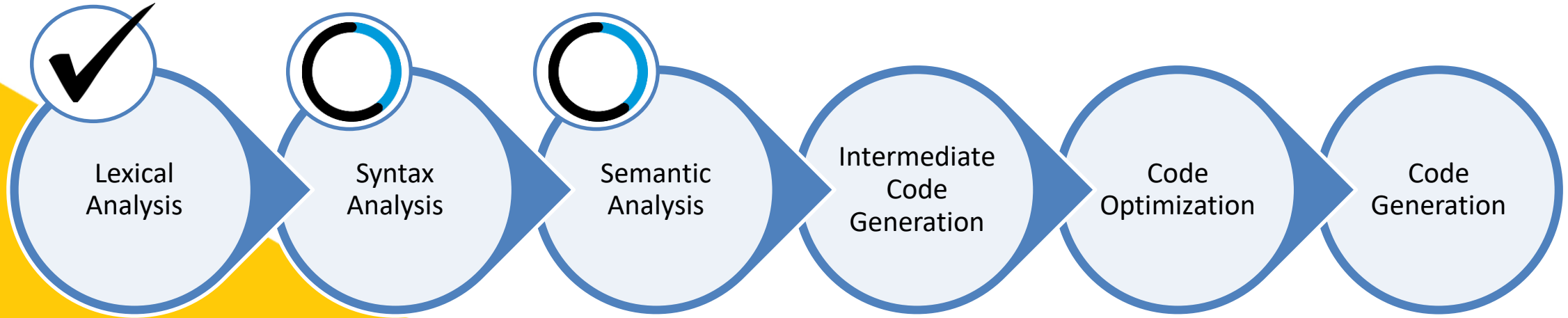
We will make our project using agile methodologies following agile values and principals.

Agile project management is a methodology that is commonly used to deliver complex projects due to its adaptiveness. It emphasizes collaboration, flexibility, continuous improvement, and high quality results.



UPES
UNIVERSITY WITH A PURPOSE

Progress:



UPES
UNIVERSITY WITH A PURPOSE

Lexical Analysis(Algorithm):

Main.c

1. Start
2. Give file input
3. Store Tokens
4. Define global variables
5. if argc != 2
   usage(argv[0]);
   A. Print out a usage if started incorrectly

6. Initialise global variables
7. Open input file
   A. Print error if unable to open properly
   B. Infile = fopen(argv[1], "r")) == NULL

8. Scan file:
   A. Define tokens: "+", "-", "*", "/", "intlit"
   B. while (scan(&T))
      a. Print token string
   C. If token == int
      Print value

9. Finish

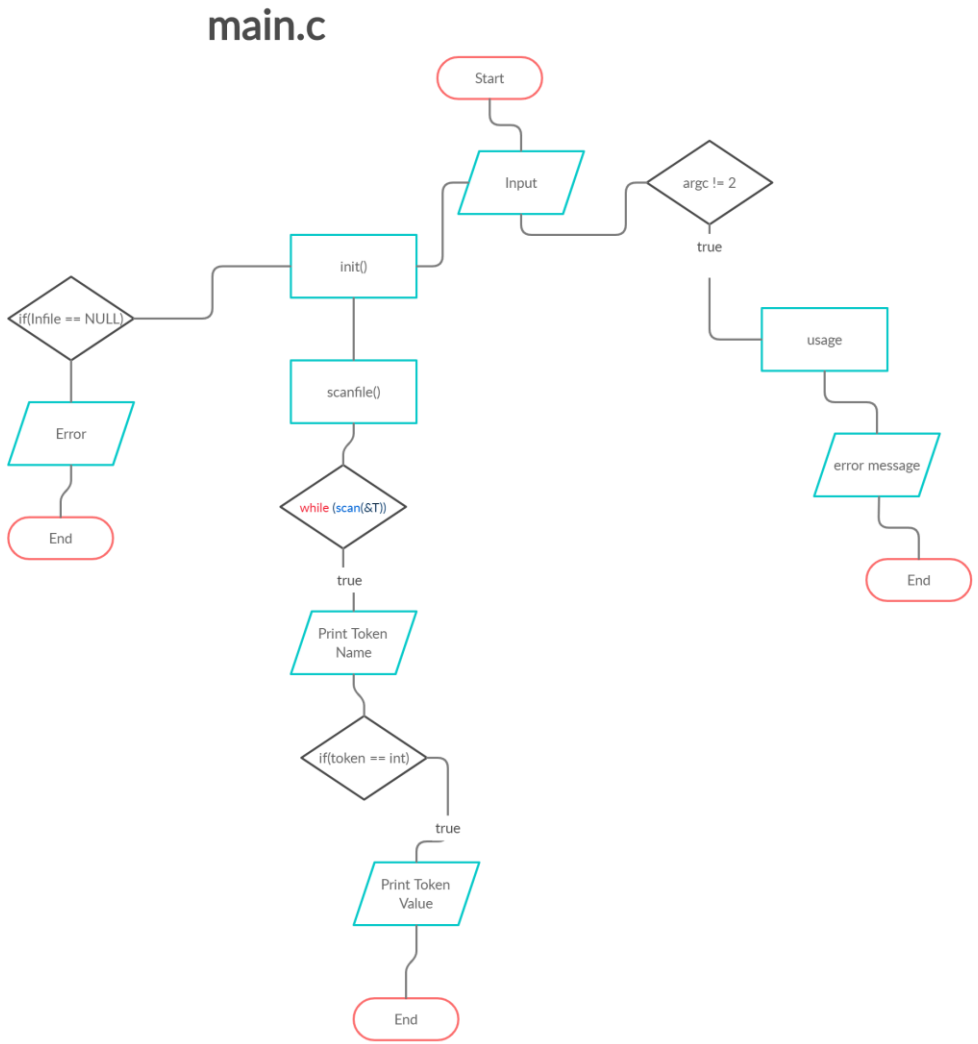
Lexical Analysis(Algorithm):

Scan.c

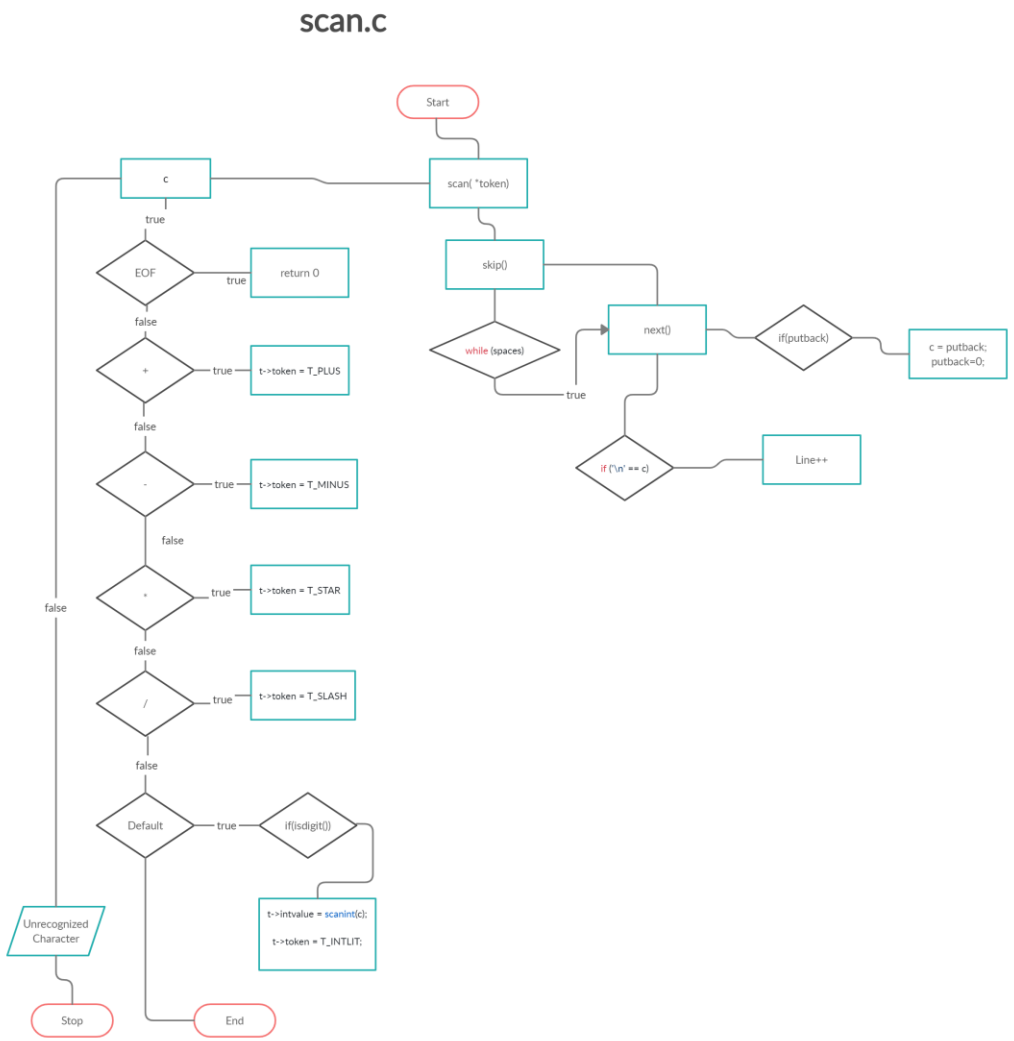
1. Define pointers
2. Define next()
    a. Define flag variable
    b. If there is a putback variable
        i. flag = Putback;
        ii. Putback = 0;
    c. Read Input file
    d. Increase line count
3. Skip unwanted variables
    a. Define flag variable
    b. Call next()
    c. While flag == ' ' or flag ==’\t’ or flag == ‘\n’ or flag == ‘\r’ or flag == ‘\f’
        i. flag = next()
4. Read tokens
    a. Define switch case:
        i. Case + : token = T_PLUS
        ii. Case - : token = T_MINUS
        iii. Case * : token = T_STAR
        iv. Case / : token = T_SLASH
        v. Default: scan integer:
            1. Define variables k, val=0
            2. Convert character to integer:
                a. While (k = chrpos("0123456789", c)) >= 0
                i. val = val * 10 + k
            3. If a non integer value is found
                a. putback()
5. Return token
6. Finish

UPES
UNIVERSITY WITH A PURPOSE

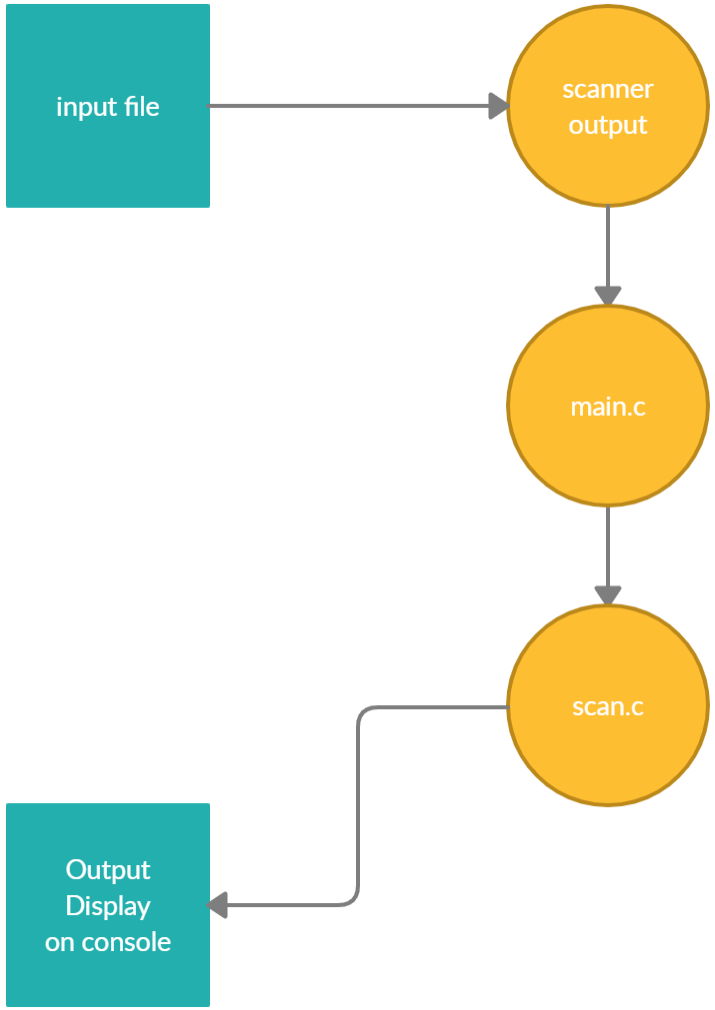
Lexical Analysis(Flowchart):



Lexical Analysis(Flowchart):



Lexical Analysis:

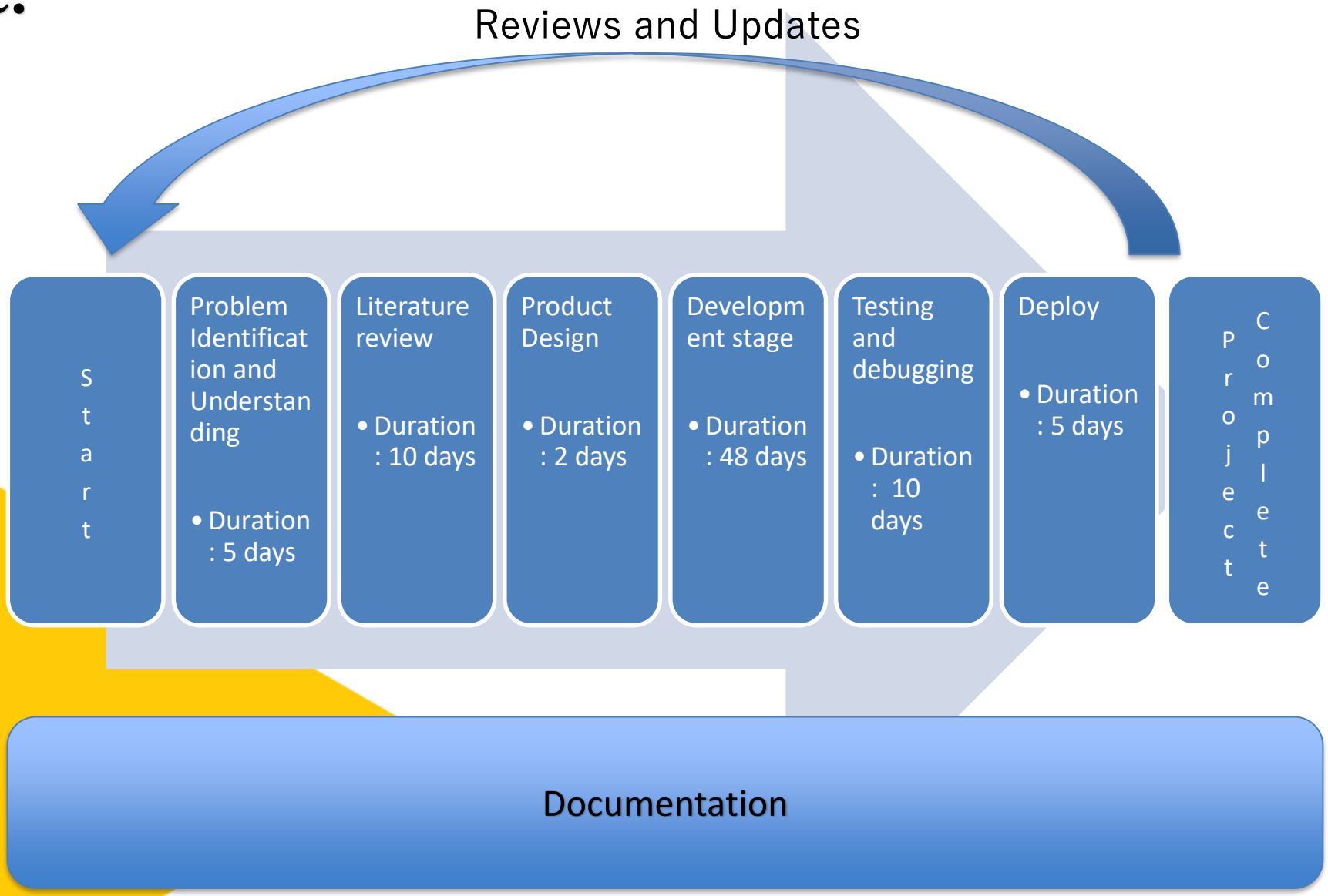


Lexical Analysis(Output):

```
neha@neha-VirtualBox:~/minor1/Web-based-compiler/Lexical Analyser$ ls
data.h  decl.h  defs.h  InputFile1  main.c  Makefile  scan.c
neha@neha-VirtualBox:~/minor1/Web-based-compiler/Lexical Analyser$ make
cc -o scanner -g main.c scan.c
neha@neha-VirtualBox:~/minor1/Web-based-compiler/Lexical Analyser$ ls
data.h  decl.h  defs.h  InputFile1  main.c  Makefile  scan.c  scanner
neha@neha-VirtualBox:~/minor1/Web-based-compiler/Lexical Analyser$ cat InputFil
e1
2
Text Editor
irtualBox:~/minor1/Web-based-compiler/Lexical Analyser$ ./scanner In
putFile1
Token intlit, value 2
Token /
Token intlit, value 3
Token *
Token intlit, value 5
Token +
Token intlit, value 7
Token -
Token intlit, value 11
neha@neha-VirtualBox:~/minor1/Web-based-compiler/Lexical Analyser$ ls
data.h  decl.h  defs.h  InputFile1  main.c  Makefile  scan.c  scanner
neha@neha-VirtualBox:~/minor1/Web-based-compiler/Lexical Analyser$ make clean
rm -f scanner *.o
neha@neha-VirtualBox:~/minor1/Web-based-compiler/Lexical Analyser$ ls
data.h  decl.h  defs.h  InputFile1  main.c  Makefile  scan.c
neha@neha-VirtualBox:~/minor1/Web-based-compiler/Lexical Analyser$
```

UPES
UNIVERSITY WITH A PURPOSE

Schedule:



UPES
UNIVERSITY WITH A PURPOSE

THANK YOU

UPES

UNIVERSITY WITH A PURPOSE