CPP

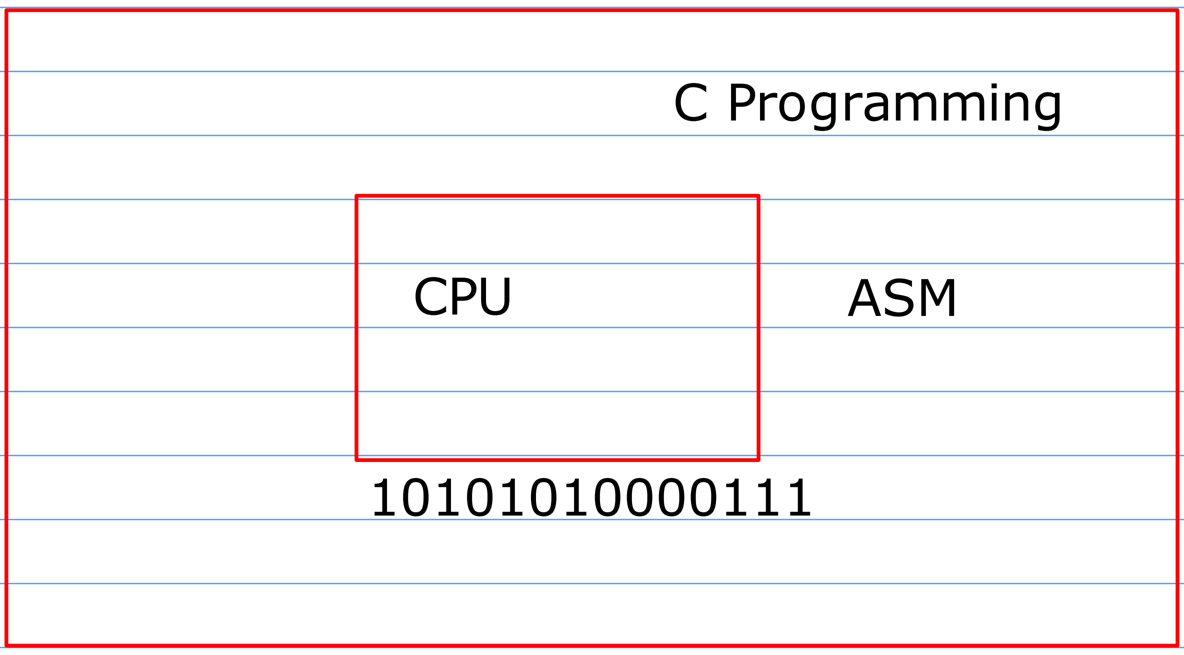
Application

- Console based application
- Desktop Application
- Web Application
- Database Application

FULL STACK DEVELOPMENT

Java

C++



C Programming

CPU

ASM

10101010000111

OOP Concepts

8-1:30
10-
12
2:15-7

OOP

- It is a methodology

1. Major Pillar
   - Abstraction
   - Encapsulation
   - Modularity
   - Hirerachy

For any OOP Language following these major pillars are mandatory

2. Minor Pillar
   - Typing/Polymorphism
   - Persistance
   - Concurrency

Optional

Abstraction

```
printf("%d");
scanf();
```

calling a function/
creating an object represents abstraction

Encapsulation

- Binding the data and code together

Defining the function/
Defining the classes represents Encapsulation

Modularity

Dividing the code into smaller modules like functions/ classes or different files

Hirerachy   Reusability

-

```
class Date{          class Employee{        class Product{          class Customer{
day,month,year
}                    }                      }                       }
```

Employee has-a Doj -> Association
Employee is-a Doj // NOT OK

```
class Person{                        class Employee{                 classs Student{
name,mobile,email,address
}
                                     }                               }
```

Employee is-a Person -> Inheritance
Employee has-a Person // NOT OK

Minor Pillars

1. Typing/ polymorphism
    - compile time
    - RunTime

```
printf("Hello");
printf("%d",num);
printf("Num2 = %d",num2);
printf("num1 = %d, num2 = %d",num1,num2);
```

2. Persistance
    - To persist the data.
    - to store the data permanantly
    - using file io / or using database connectivity

3. Concurrency
    - Performing multiple tasks at the same time.
    - Multithreading represents concurrency

OOSD - Object Oriented Software Development

1. OOA -> Object Oriented Analysis
2. OOD -> Object Oriented Design
3. OOP -> Object Oriented Programming

Attendance System OOSD

OOA-

Student,Employee,Attendance

OOD -

```
class Student{
rollno,
name,
course,
marks,

}
```

```
class Time{
hr;
min;
}

class Date{
day
month
year
}
```

```
class Employee{
empid,
name,
dept,
salary,

}
```

```
class Attendance{
type - E/S
id
inpunch
outpunch
punch_date

}
```

OOP

CPP

C and Simula

class into C Programming Language

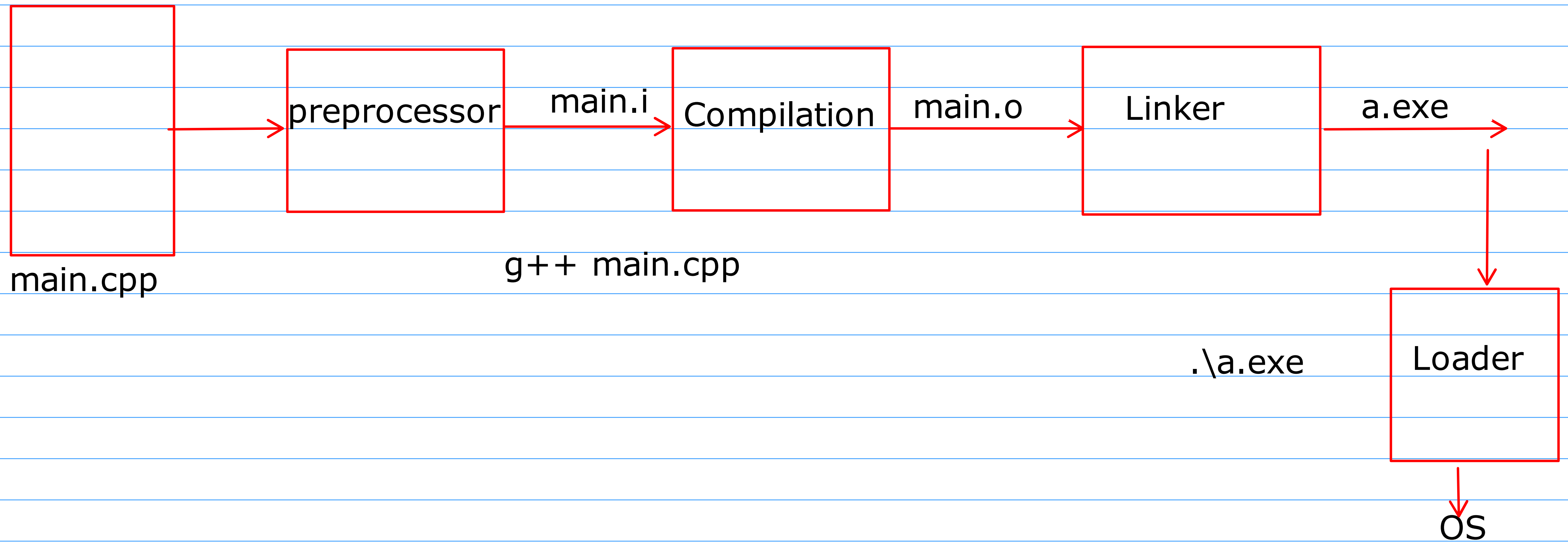
C++ CPP

Compiler

Editor- vscode

IDE

Flow of Execution



g++ main.cpp

.\a.exe