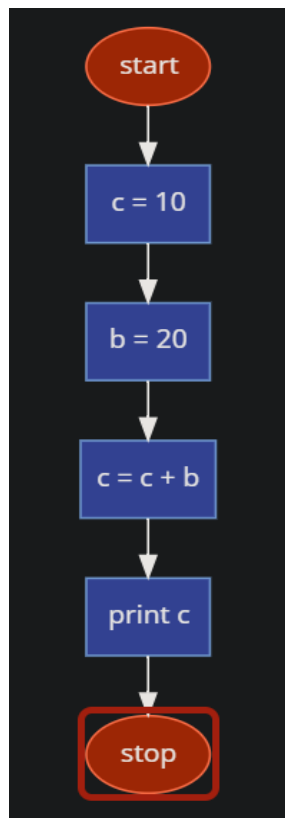


## Sequence

- Programs are loaded in an order in the memory, each are same in size and has a unique address. By default, every program runs in a sequence unless if we add any conditions or loops to jump into other programs. The program calls one instruction by its address and then it loads the next instruction by its address and runs this happen continuously. Sequence is simply a default step by step execution of statement in the program from top to bottom.

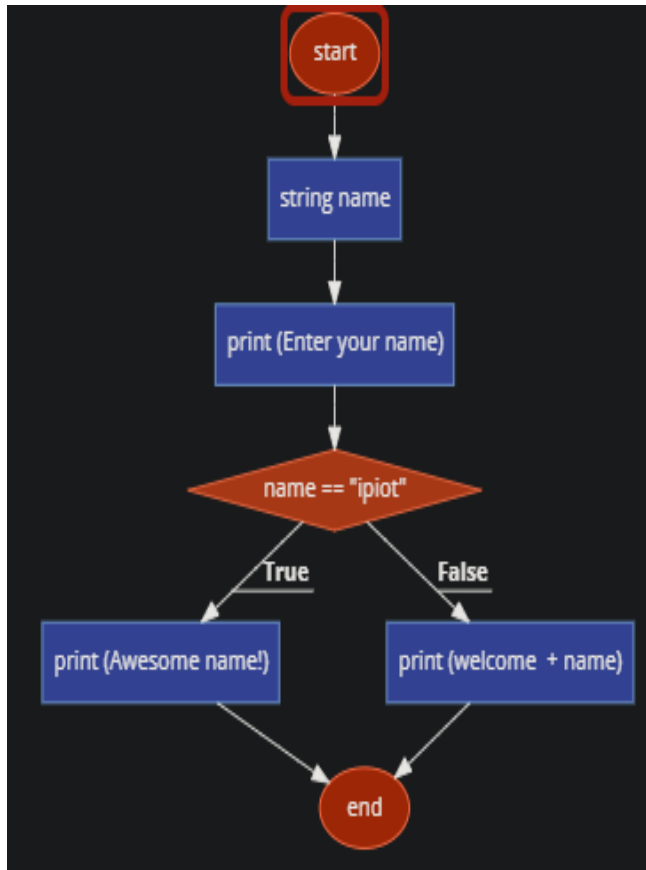


```
int b, c;  
c = 10;  
b = 20;  
c = c + b;  
write_line (c);  
return 0;
```

- Here we had declared the “int main” it denotes that it is the main function, and we declare two integer variable b, c. simply we created two boxes named b and c. we then assign the box named c (here it’s the variable c) a value of 10 and the box named b a value of 20. Afterwards we had declared that “c = c + b” where it says as the c box is equal to the value addition of c and b stores that addition value in the c box cancelling the old value. Then we ask to print it so that it gives an output as “c is 30”. This how a program runs in a sequence (step by step process).

## Selection (if , case)

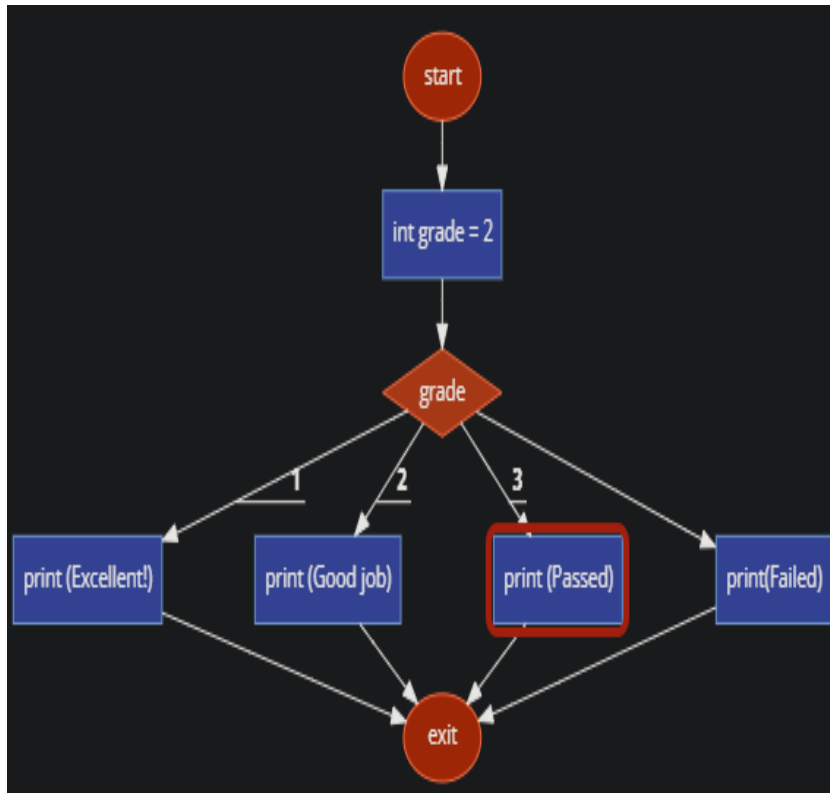
- Selection is used programs to conditionally execute various code blocks based on the condition applied. If statement is a branching into two allowing to select the value based on Boolean expression (as true or false). The statement has two branches where the program will select one if the condition is true and it will select the other branch if the statement is false. Even thou it has two branches the program has only one exit.
- Below is a code block to show how the if statement is performed.



```
string name;
print (Enter your name)
if (name == "IPIOT")
    print (Awesome name!)
else
    print (welcome + name)
end
```

- In this program there is a variable created as name, and then it prints to 'enter the name' (this is performed to get an input from the user). Once a name is entered it is going to the if statement the condition which I have applied there, I says (if the name is equal to ipiot) the if statement would select the true branch and print saying, 'awesome name!' if the name entered was not equal to the name ipiot as in the if condition the statement going to select the false branch and print 'welcome (with the name entered)'. Then the program exits.

- Case statement is also a branching statement which allows us to create many paths to be executed based on the matching value. Each path will have a value and it will be determined when we run the code and based on the value it would select the matching path. Case statement is bit different from if conditions where case statement has many paths but still it also has only one exit.
- Below is a code block to explain how the case statement is performed.



```
int grade = 2;
```

```
switch(grade)
```

```
{
```

```
  case 1:
```

```
    print (Excellent!);
```

```
    break;
```

```
  case 2:
```

```
    print (Good job);
```

```
    break;
```

```
  case 3:
```

```
    print (Passed);
```

```
    break;
```

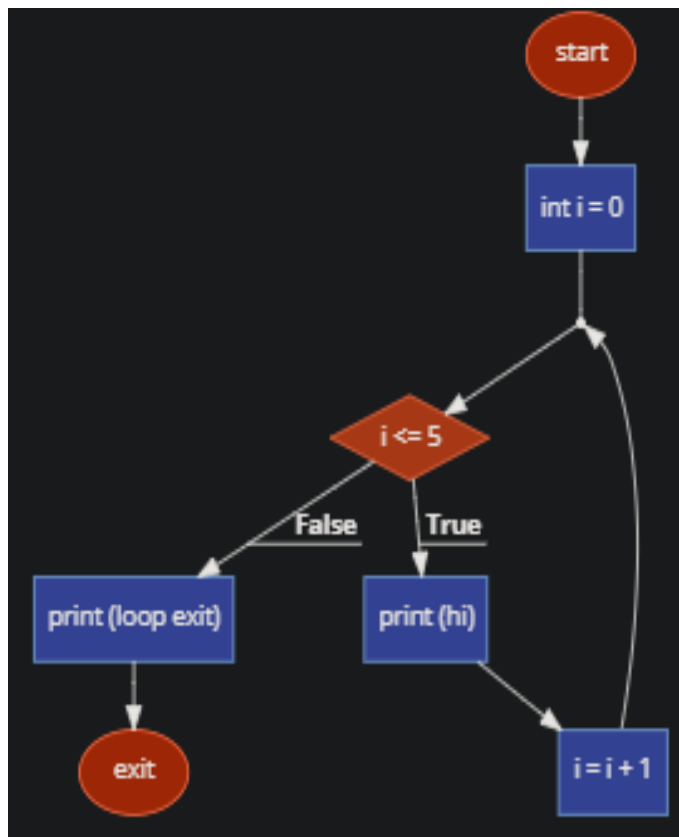
```
  default:
```

```
    print (Failed)
```

- Here in the case statement it allows to execute different code block based on the value of the variable. When the program reaches the switch statement it checks the value of the grade, for example if it is 1 and the value 1 matches the case 1 statement it executes the code in case 1 as the same happens for the others based on the value. If the value does not match any of the cases, then it executes the default. The break in the case statement is very useful because they exits the switch statement after the path is completed if it is not there it would keep executing the other paths also.

## Repetition (While, do ... while)

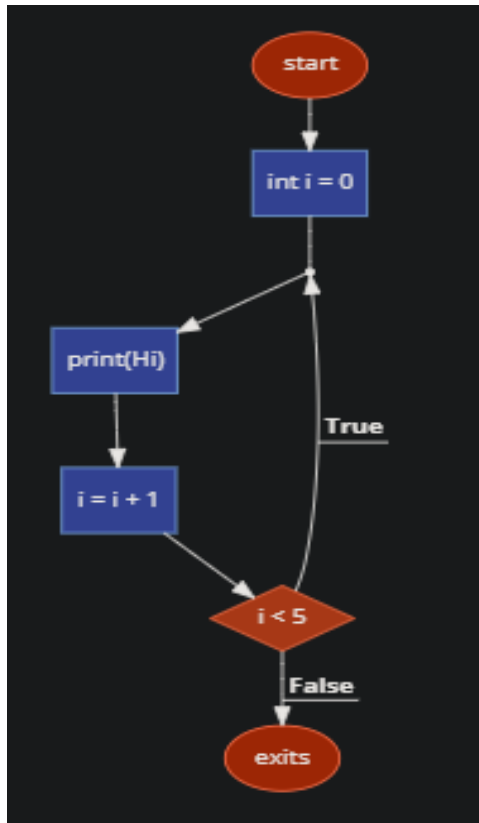
- While loop executes the code block repeatedly and checks if the condition is true. if it is not true the condition is repeated again and again till it is true, it checks the condition after each looping iteration. In while loop it first checks the condition and if it is true it loops all the time it is true till it becomes false.



```
start
{
  int i = 0;
  while (i <= 5) {
    print (hi);
    i = i + 1;
  }
  print (loop exit);
  exit;
}
```

- A integer variable *i* is been declared and a value of zero is being assigned. The code enters the while loop where this will repeated the code inside it till it is true, the condition in it is, if *i* is less than or equal to 5. In this case it will loop till the *i* is becomes below 5 or till it becomes equal to it. It sides the loop it prints 'hi' each time it loops, it will increase one each time it loops. It will be incrementing till it becomes 6, once it is 6 the condition becomes false and exits the loop by printing 'loop exit'.

- Do-while loop is similar to the while loop but the difference in do-while is it executes the code for the first time and then checks the condition as in while loop it was checking the condition and executing the code. Do-while at least runs once when it is executed other than a while loop checks its condition before it runs.



```
start
int i = 0;
do {
    print(Hi);
    i = i + 1;
} while (i < 5);
exits
```

- This starts a do-while loop and once it executed it will print 'hi' first before checking the condition and add 1 to i, then it checks the condition is i<5 (is i less than 5) the statement becomes true and it repeats and print 'hi' again. The same process repeats till i<5 condition becomes false.