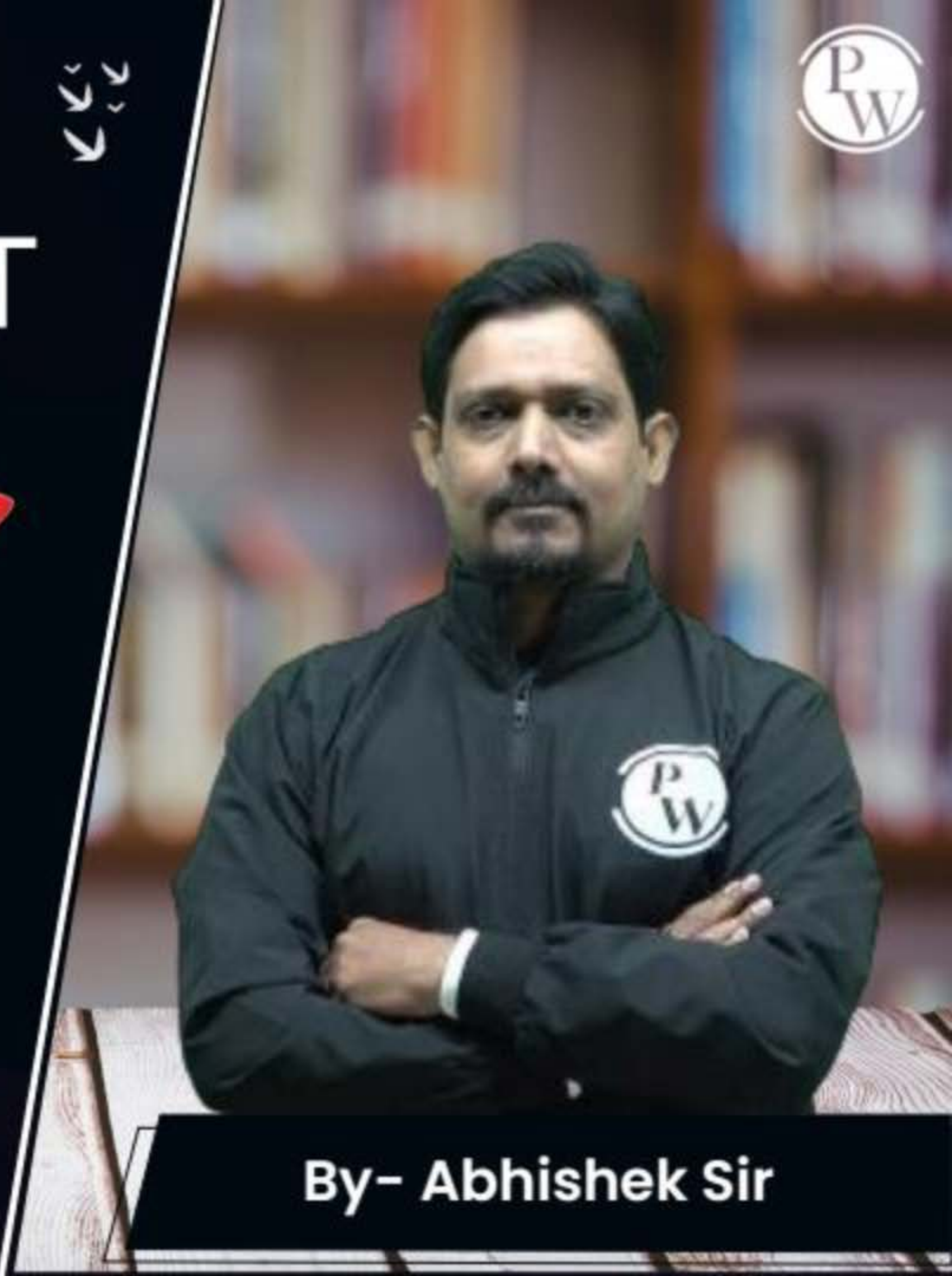
# Computer Science & IT

## C Programming

**Control Flow Statement**

**Lecture No. 01**



**By- Abhishek Sir**

# Recap of Previous Lecture

**Topic** Shift left <<, shift right >>

**Topic** printf return value

**Topic** Ternary operator

**Topic** Comma,

**Topic**

Slide

# Topics to be Covered

Topic if statement conditional

Topic Switch statement

Topic for Loop

Topic

Topic

Slide

# [GATE-2022]

#Q. What is printed by the following ANSI C program?

```
#include<stdio.h>
int main(int argc, char *argv[]){
    char a = 'P';
    char b = 'x';
    char c = (a & b) + '*';
    char d = (a | b) - '-';
    char e = (a ^ b) + '+';
    printf("%c %c %c\n", c, d, e);
    return 0;
}
```

ASCII encoding for relevant characters is given below

| A | B | C | ... | Z |
|---|---|---|---|---|
| 65 | 66 | 67 | ... | 90 |

| a | b | c | ... | z |
|---|---|---|---|---|
| 97 | 98 | 99 | ... | 122 |

| * | + | – |
|---|---|---|
| 42 | 43 | 45 |

[✓ (A) z K S]

(B) 122 75 83

(C) * - +

(D) P x +

# [GATE-2022]

A - 65  Z - 90

a - 97  z - 122

#Q. What is printed by the following ANSI C program?

```
#include<stdio.h>
int main(int argc, char *argv[]){
    char a = 'P';    80
    char b = 'x';   120
    char c = (a & b) + '*';
    char d = (a | b) - '-';
    char e = (a ^ b) + '+';
    printf("%c %c %c\n", c, d, e);
    return 0;
}
```

ASCII encoding for relevant chara

A B C D E F G H I J K L M N O P Q R S T

U V W X Y Z

a = 01010000  b = 01111000

$$
\begin{array}{r}
b = 01111000 \\
\hline
01010000
\end{array}
$$

$$
\begin{array}{r}
80 \\
\hline
+ \quad 42 \\
\hline
122
\end{array}
$$

# [GATE-2022]

A - 65   Z - 90

a - 97   z - 122

#Q. What is printed by the following ANSI C program?

```
#include<stdio.h>
int main(int argc, char *argv[]){
    char a = 'P';    80
    char b = 'x';   120
    char c = (a & b) + '*';
    char d = (a | b) - '-';
    char e = (a ^ b) + '+';
    printf("%c %c %c\n", c, d, e);
    return 0;
}
```

ASCII encoding for relevant chara

A B C D E F G H I J K L M N O P Q R S T

U V W X Y Z

a = 01010000   b = 01111000

b = 01111000

01111000   120

$120 - 45 = 75$ $\textcircled{K}$

# Toipc: Question

```
#include<stdio.h>
int x = 40;
int  main() {
        int x = 30;
       {
            int x = 20;
           {
                  int x = 10;
                 printf("%d", x);
           }
         }
       return 0 ;
}
```

10

Scope of variable { — function

} if

Loop

Block : visibility and Life time

Slide

# Toipc: Question

```
#include<stdio.h>
int x = 40;
int  main() {
        int x = 30;
       {
            int x = 20;
            {
                //int x = 10;
                 printf("%d", x); 20
            }
          }
       return 0 ;
}
```

Scope of variable { — function
} if
Loop

Block : visibility and Life time

undeclared and undefined variable

Static scoping - C follow

Outside of bracket for finding value of undeclared undefind variable

Slide

# Toipc: Question

```
#include<stdio.h>
int x = 40;
int  main() {
       int x = 30;
      {
        //int x = 20;
         {
           //int x = 10;
            printf("%d", x); 30
         }
       }
      return 0 ;
}
```

Scope of variable { — function
} if
Loop

Block : visibility and Life time

undeclared and undefined variable

Static scoping - C follow

Outside of bracket for finding value of undeclared undefind variable

Slide

# Toipc: Question

```
#include<stdio.h>
int x = 40;
int  main() {
     //int x = 30;
      {
         //int x = 20;
          {
              //int x = 10;
               printf("%d", x); 40
          }
        }
      return 0 ;
}
```

Scope of variable { — function
} if

Loop

Block : visibility and Life time

undeclared and undefined variable

Static scoping - C follow

Outside of bracket for finding value of undeclared undefind variable

Slide

# Toipc: Question

```
#include<stdio.h>
//int x = 40;
int  main() {
     //int x = 30;
      {
        //int x = 20;
         {
           //int x = 10;
           printf("%d", x);
         }
      }
     return 0 ;
}
```

variable x not declared.

Scope of variable { — function
} if
Loop

Block : visibility and Life time

undeclared and undefined variable

Static scoping - C follow

Outside of bracket for finding value of undeclared undefind variable

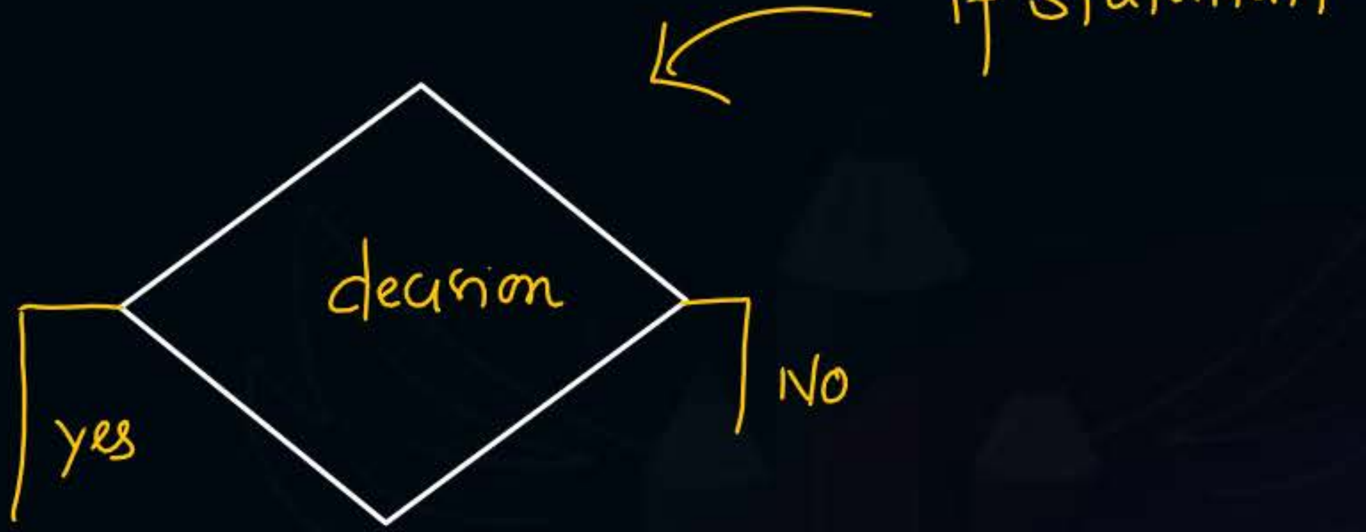
Slide

# Control Flow Statement

1 Selection, branching/conditional/ Decision

2 Iterative statement. for, while, do while

3 Jump statements



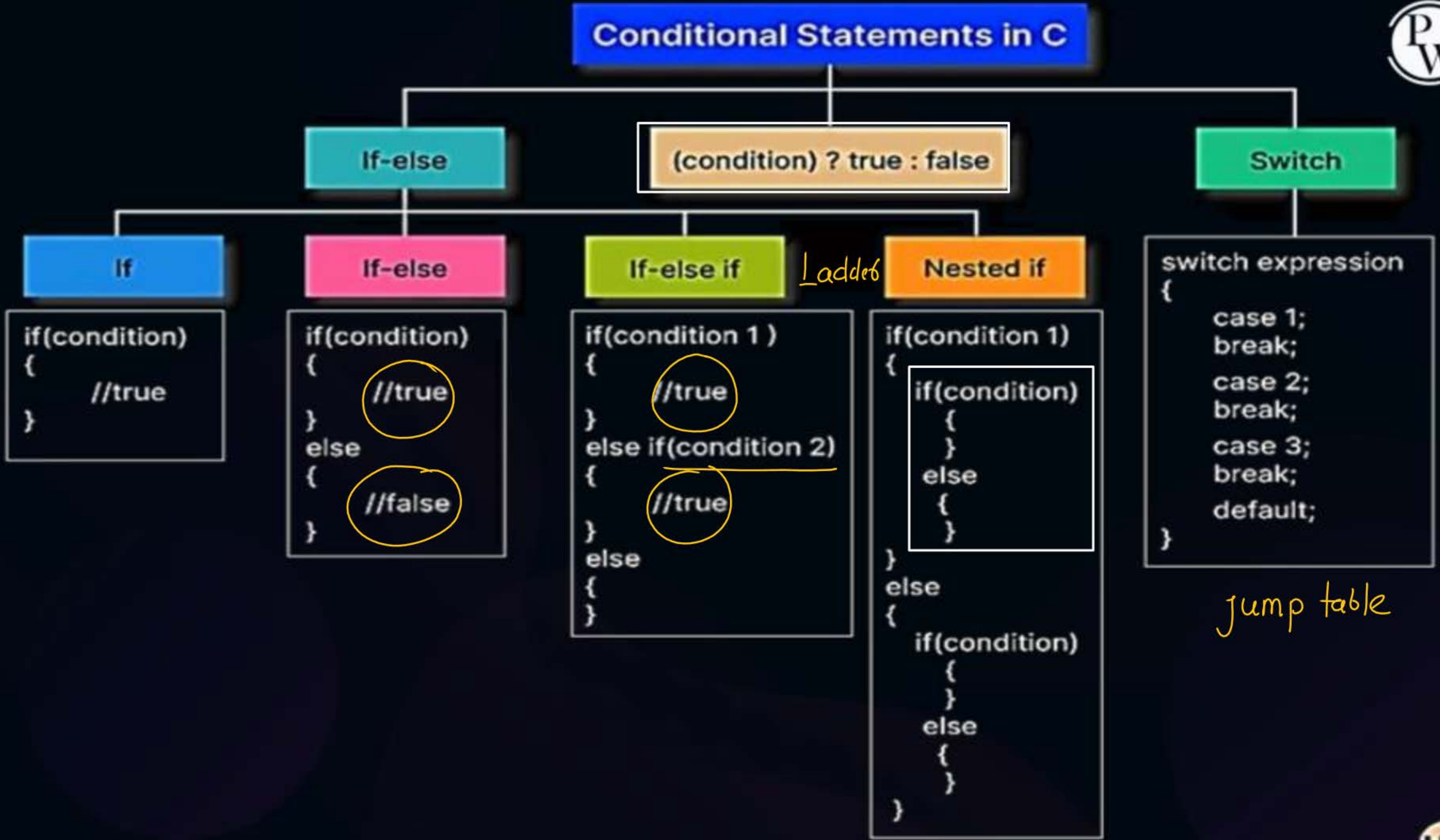
Slide

# Types of Control Flow statement

- Selection or Branching or Decision or Conditional
- Iterative statement
- Jump statements

Slide

# Conditional Statements in C



```
if(condition)
{
        //true
}
```

```
if(condition)
{
        //true
}
else
{
        //false
}
```

```
if(condition 1 )
{
        //true
}
else if(condition 2)
{
        //true
}
else
{
}
```

```
if(condition 1)
{
  if(condition)
    {
    }
  else
    {
    }
}
else
{
  if(condition)
    {
    }
  else
    {
    }
}
```

```
switch expression
{
        case 1;
        break;
        case 2;
        break;
        case 3;
        break;
        default;
}
```

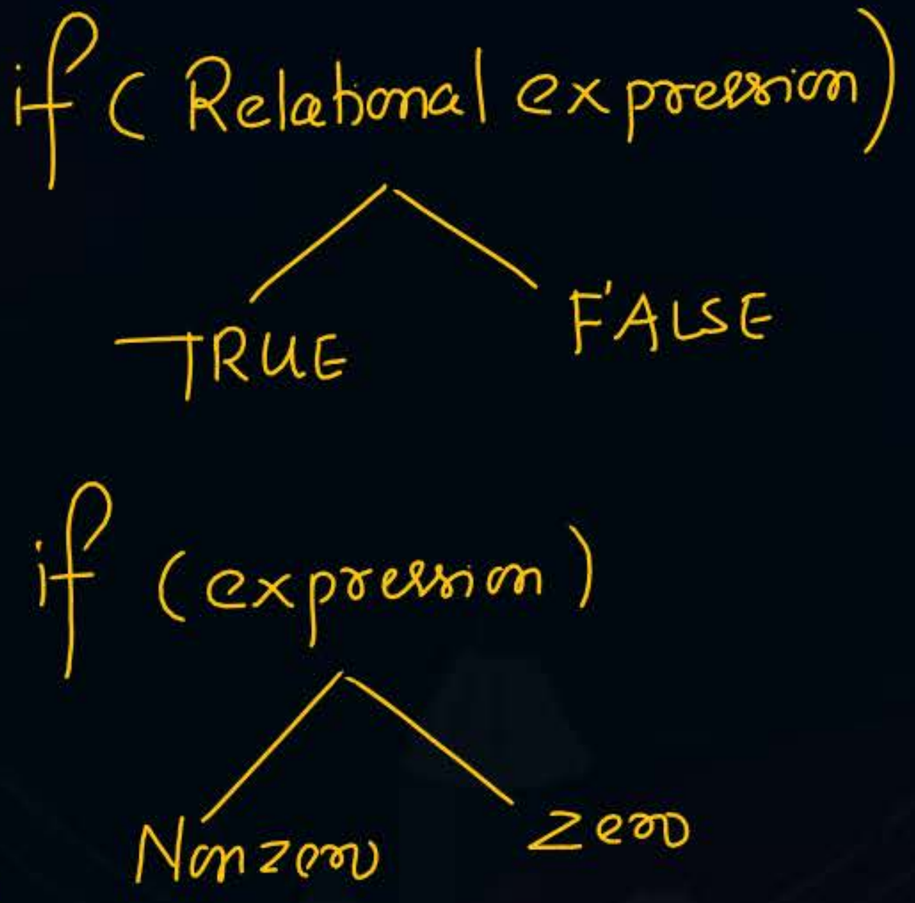
jump table

# Question

```
#include <stdio.h>
int main (){

        int i=0;
        if(i++){
            printf(“Namaskar");
        }
        printf(“\t Dunia");
        return 0;
}
```

if(0)


if( Relational expression)

TRUE FALSE

if (expression)

Nonzero zero

Dunia

printf("%d", i) = //1

(A) NamaskarDunia (B) Namaskar Dunia

(C) Namaskar [✓ (D) Dunia]

Slide

# Question

```
#include <stdio.h>
int main (){

        int a=0;
        if(a++);
        {
                a = a+10;
                a = a-5       ;

        }
        printf("%d", a);
        return 0;
}
```

if(a++); ← block selection power is gone

↑
a++ execute

a | 1 | 11 | 6

(A) 5 [✓ (B) 6] (C) 15 (D) 20

Slide

```
int a=0,

if(a++); {

    a =a+10,
}
else  {

      a = a-5,
}
printf("%d", a),
```

if(a++);  Terminate

else without if

Error

# if statement Example

```
#include <stdio.h>
     int main (){
          int a = 10
          if(a+2) {
               printf("Namaskar" );
          }
          printf(“Dosto");
          return 0;
}
```

if(a+2) = if(12)

Nonzero - TRUE

Namaskar Dosto

Slide

# if statement Example

```
#include <stdio.h>
      int main (){
            int a = 10
            if(a-10) {
                  printf(“Namaskar" );
            }
            printf(“Dosto");
            return 0;
}
```

if(0)

Dosto

Slide

# If else ladder

```
#include <stdio.h>
int foo(int n1,int  n2,int n3){
    if (n1 >= n2 && n1 >= n3) {
       return  n1;
    } else if (n2 >= n1 && n2 >= n3) {
        return  n2;✓
    } else {
        return  n3;
    }
}
int main() {
    printf("%d", foo(12, 70, 40));
    return 0;
}
```

Output of the program is________

$n_1 = 12$, $n_2$ 70, $n_3 = 40$

$\boxed{12 >= 70}$ && 12<=40

} else if (n2 >= n1 && n2 >= n3) { → $\boxed{70>=12 \;\&\&\; 70>=40}$

finding max of 3 Number.

Slide

Left to right Associativity

$$\boxed{a = 10, 20, 30;}$$

14 ↑ 15

$a = 10$ preredence

$$a = (\underline{10}, \underline{20}, \underline{30})$$

↑ keep

$$\boxed{a = 30}$$

# Question

Consider the following program:

```
#include<stdio.h>
int main(){
    int a=19, b=20;
    if(a^b<b|a)
        printf("%d",a+++--b>>2);
    else
        printf("%d", ++a+--b<<2);
    return 0;
}
```

The output is

$$\begin{array}{r} 1011 \\ 1011 \\ \hline \end{array}$$

$$\begin{array}{r} 10011 \\ 00000 \\ \hline 10011 \\ 10011 \\ \hline \end{array}$$

$a \wedge \underbrace{b < b}_{} | a$

$a \wedge 0 | a$

$a{+}{+} + {-}{-}b >> 2$

$19 + 19$

$38 >> 2 = \frac{38}{4} = \underline{9}$

Slide

# Question

Consider the following program:

```
#include<stdio.h>
int main(){
    int a=19, b=20;
    if(a&b?1?0:0:1)
        printf("%d",a++++);
    else
        printf("%d", ++a+--b<<2);
    return 0;
} The output is
```

(A) 156

[✓ (B) Error]

(C) 38

(D) 40

$a\&b?\boxed{1?0:0}:1$

$a\&b?0:1$

if(0)

$$\begin{array}{r} 10011 \\ 10100 \\ \hline 10000 \\ \hline \end{array}$$

20+19

39×4

156

3

a++++ wrong

(a++)++

expression/constant ++ Not applied

Slide

Q. What is the output of the following program?

ISRO

```
#include <stdio.h>
int main () {
  int x = 2, y = 5;
  if (x < y) return (x = x + y);
   else printf("z1");
  printf("z2");
  return 0;
}
```

if(2< 5) return x=x+y
TRUE x=7

7 return ← Terminate

OS return 0 ←

Successful completion

(a) z2

(b) z1z2

(c) 7Z2

[✓ (d) None of the above]

#Q Consider the following program fragment

```
if (a > b) if (b >c) s1 ; else s2;
```

s2 will be executed if

(a) a < = b

(b) b > c

(c) b >= c and a <= b

(d) a > b and b <=c

## Dangling else

$\boxed{a = a++;}$

Sequence point

Compiler Design

```
if(a>b)
      if(b>c)
            s1;
else
         s2
X
```

```
if (a>b) ✓
      if(b>c)
            s1
      else
            s2
```

else will be part of immediate if

# Switch Syntax

```
switch(expression){
   case constant-expression  :
      statement(s);
      break; //optional
   case constant-expression  :
      statement(s);
      break; //optional

   // you can have any number of case statements.
   default : //Optional
      statement(s);
}
```

Not constant error

* Switch statement allows a expression to be tested for equality against list of values.

* break takes control outside of switch block

* if No case matches the default case executes

* position of default does not matter

* Break and default both are ophonal

# Question

```
#include <stdio.h>
     int main(){
      int a = 5;
       switch(a){
               case 4: printf("%d",5);
               case 5: printf("%d",5);
               case 6: printf("%d",6);
       }
     }
```

output :

5 6

if No break is present the all subsequent cases evaluates TRUE

# Question

```
#include <stdio.h>
     int main(){
      int a = 5;
       switch(a-1){
               case a: printf("%d",5);
               case 5: printf("%d",5);
               case 6: printf("%d",6);
      }
     }
```

must be

constant expression

variable | Not allowed

output:

Error

Reason

case expression

is a variable

# Question

```
#include <stdio.h>
int main(){


    switch(13/4){
        case 4: printf("%d",4); break ;
        case 3: printf("%d",2); break ;
        case 5: printf("%d",5); break ;
}
    }
```

output 2

$13/4 \Rightarrow 3$

case 3:

# switch(expression)

```
#include <stdio.h>
int main(){
        switch(13/4.0){
        case 3: printf("%d",4); break ;
        case 4: printf("%d",2); break ;
        case 5: printf("%d",5); break ;
        }
    }
```

float

only integer allowed

## Output

Error

## Reason

$13/4.0 = 3.25$

↑ float

Swith expression must be an integer

# Position of default does not matter

default is optional and can be placed any where in Switch block

# switch(expression)

char ASCII

BASCII

'a' - 97

'b' - 98

'c' - 99

'd' - 100

```
#include <stdio.h>
int main(){
      switch('d'){
       case 'a': printf("%d",4); break ;
       case 'b': printf("%d",2); break ;
       default: printf("None");
       case 'c': printf("%d",5); break ;
      }
}
```

## Output

None5

## Reason

'a', 'b' 'c', 'd'

are integer

ASCII value

# switch(expression)

```
#include <stdio.h>
int main(){
        switch(4){
        case 2: printf("%d",4); break ;
        case 1+1: printf("%d",2); break ;
        default: printf("None");
     }
}
```

Output

Error

Reason

Case 2:

Case 1+1: case 2

duplicate case
Not allowed

# Question

```c
#include <stdio.h>
int main(){
   int a = 80;
   switch(-12%45+36/9/2*16+60){
      case 80: a = a+10;
      case 5: a++;
      default : a = a>>2;

   }
   printf("%d", a) ;

}
```

(A) 20

(B) 21

[✓ (C) 22]

(D) 23

$\underline{-12\%45} + \underbrace{36/9}/2 * 16 + 60$

$-12 + 32 + 60$

$= 20+60 = \boxed{80}$

$a = 90$

$a = 91$

$a = a>>2 = \left\lfloor \frac{91}{4} \right\rfloor = 22$

Slide

# Question

```
#include <stdio.h>
int main(){
   int a = 5, b=24,c= 10;
   switch(28/5) {
      case 5:     a = a&b;
      case 6:     b = b^c; break;
      default:    a = a>>2;
   }
   printf("%d",a+b+c);

}
```

$\frac{28}{5} = 5$

a 00101

b: 11000

00000

a = 0

b = 18

b: 11000

c. 01010

10010 (18)

0+18+10 = 28

(A) 20

(B) 21

(C) 22

[✓ (D) 28]

Slide

# 2 mins Summary

Topic if - else

Topic Switch

Topic

Topic

Topic

Slide

# THANK - YOU