

Ch. 3 = Conditional Instructions

Sometimes we watch comedy videos on YouTube if the day is Sunday.

Sometimes we order junk food if it is our friend's birthday in the hostel.

You might want to buy an umbrella if it's raining & you have the money.

All these are decisions which depend on a condition being met.

In C language too, we must be able to execute instructions on a condition (if) being met.

Decision making Instructions in C

- If-else statement
- Switch statement

If-else Statement

The syntax of an If-else statement in C looks like

If (Condition to be checked)

Statements - If-condition - true ;

else if

Statement - If-Condition- false ;

for e.g.

```
#include <stdio.h>
```

```
int main() {
```

```
    int a, b;
```

```
    printf("Enter a number\n");
```

```
    scanf("%d", &a);
```

```
    if (a % 2 == 0) {
```

```
        printf("%d is even\n", a);
```

```
    } else {
```

```
        printf("%d is odd\n", a);
```

```
    } return 0;
```

Ans:

Enter a number

For e.g.) 55

* 55 is odd

* 55 is odd

=> 55 is odd

e.g. 2

int a = 23

if (a > 18) {

printf("You scanned a large number\n");

Note that else block is not necessary but optional

Relational Operators in C

Relational operators are used to evaluate conditions (true or false) inside the if statements. Some examples of relational operators are

$= = \rightarrow$ Equals to

$< \rightarrow$ Less than

$<= \rightarrow$ Less than or equal to

$> \rightarrow$ Greater than

$>= \rightarrow$ Greater than or equal to

$\neq \rightarrow$ Not equals to

Imp. note $\rightarrow '='$ is used for assignment where as $'=='$ is used for equality check.

The condition can be any valid expression. In C a non-zero value is considered to be true.

Logical Operators

$\&\&$ → Logical AND (true only if both the operands are true)

$\|$ → Logical OR (true if one operand is true)

! → Logical NOT (negate the operand)

They are used to provide logical tests in programs

for E.g.

```
#include<stdio.h>
```

```
int main()
```

```
{ int age;
```

```
printf("Enter your age\n");
```

```
scanf("%d", &age);
```

```
if(age <= 70 && age >= 18)
```

```
    printf("You can drive\n");
```

```
else
```

```
    printf("You cannot drive\n");
```

return 0;
}

Ans:

Enter your age

For. (i) 17

(ii) 18

You can drive

(i) You cannot drive

(ii) You can drive

As the number of conditions increases, the level of indentation increases. This reduces readability. Logical operators come to rescue in such cases.

else if clause

Else if Ladder

Instead of using multiple if statements, we can also use else if along with if thus forming an if-else-if-else ladder.

For e.g.

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int num;
```

```
    printf("Enter your number \n");
```

```
    scanf("%d", &num);
```

```
if (num == 1) {  
    printf("Your number is 1\n");  
}
```

```
else if (num == 2) {  
    printf("Your number is 2\n");  
}
```

```
else if (num == 3) {  
    printf("Your number is 3\n");  
}
```

```
else {  
    printf("It's not 1, 2 or 3\n");  
}
```

```
return 0;
```

(i) 1

(ii) 2

(iii) 24

Your number is 1

2

It's not 1, 2 or 3

Else of structure

if {
 // statements ;
}

else if {
}

else if {
}

else {
}

Using if-else-if-else reduces indent.
The last else is optional, also there can
be any number of 'else if'

Last else is executed only if all conditions
fail.

Type	Size	Range	Type (32) float	App. range $10^{-37} - 10^{38}$	Sig. dig. 6
int	2	-32767 - 32768	Double (64)	$10^{-307} - 10^{308}$	15
				End of table	

Operator Precedence

Priority	Operator
1 st	
2 nd	*
3 rd	/, %
4 th	+, -
5 th	<, >, <=, >=
6 th	==, !=
7 th	&&
8 th	
	=

Working from left to right

Conditional Operators

A short hand "if-else" can be written using the conditional or ternary operators.

Condition? expression-if-true : expression-if-false

Ternary operators

For e.g. `int val = (val > 5) ? 10 : 20;`

#include <stdio.h>

int main()

int a;

printf("Enter a\n");

scanf("%d", &a);

(a < 5)? printf("A is less than 5"):

printf("A is not less than 5");

`return 0;`

Ans:

Enter a

- (i) 1
 (ii) 7

- (i) A is less than 5
(ii) A is not less than 5

Switch Case Control

Instruction

Switch-Case is used when we have to make a choice i.e. "number of alternatives" for a given variable.

Switch (integer-expression)

Case C1:

Code;

break;

$E, C1, C2, C3 \rightarrow \text{Const.}$

Code: any valid C code

Case C2:

Code;

break;

B

Case C3:

Code;

break;

default:

Code 3

break;

3

The value of integer-expression is matched against C1, C2, C3... If it matches any of these cases, that case along with all subsequent "case" and "default" statements are executed

#include <stdio.h>

int main () {

int rateng;

printf("Enter your rating (1-3)\n");

scanf("%d", &rateng);

switch(rateng) {

Case 1:

printf("Your rating is 1\n");

break;

Case 2:

printf("Your rating is 2\n");

break;

Case 3:

printf("Your rating is 3\n");

break;

default:
printf("Invalid rating!\n");
break;

3

~~return 0;~~

3

Enter your rating (1-3)

3 17

①

② 4

① Your rating is 1

② Invalid rating!

Gmp. Notes

- 1) We can use switch-case statements even by writing cases in any order of our choice (not necessarily ascending)
- 2) char values are allowed as they can be easily evaluated to an integer
- 3) A switch can occur within another but in practice this is rarely done.

Practice set: Ch-3

- Q1 What will be the output of the program?
- ```
#include <stdio.h>
int a=10;
if(a==11){ printf("I am 11"); }
```

```
else {
 printf("I am not 11");
```

Ans:

I am not 11 int<sup>o</sup> = & a assignment operator

Q2 Write a program to find out whether a student is pass or fail, if it requires total 40% & at least 33% in each subject to pass. Assume 3 subjects & take marks as an input from the user.

Code: #include <stdio.h>

int main() {

int physics, chemistry, maths;

float total;

printf("Enter physics marks \n");

scanf("%d", &physics);

printf("Enter chemistry marks \n");

scanf("%d", &chemistry);

printf("Enter maths marks \n");

scanf("%d", &maths);

total = (physics + chemistry + maths) / 3;

if((total < 40) || physics < 33 || maths < 33 || chemistry <

```
{ printf("Your total percentage is %f &
you are failed", total);
```

else if

```
printf("Your total percentage is %f &
you are passed", total);
```

return 0;

Ans:

Enter Physics Marks

1

Enter Chemistry Marks

2

Enter Maths Marks

3

Your total percentage is 2 & you are fail

Q3) Calculate the income tax charged by an employee to the government as per the slates mentioned below:

Income & Tax

Tax

(0.5 - 5)L 5%

(S - 10)L

20%

Above 10L

30%

Note that there is no tax below 2.5L Take income amount as an input from the user

000,02,8 (i) (ii)

Code: #include <stdio.h>

int main() {

float tax = 0, income;

printf("Enter your income\n");

scanf("%f", &income);

If (income) >= 250000 && income <= 500000 {

15.11

tax = tax + 0.05 \* (income - 2,50,000);

y

If (income) >= 500000 && income <= 1000000 )

d

tax = tax + 0.20 \* (income - 500000);

y

If (income) >= 1000000 )

d

tax = tax + 0.30 \* (income - 1000000);

y

printf("Your net income tax to be paid  
as %f \n", tax);

return 0;

y

Enter your income

- (i) 2,50,000  
 (ii) 3,00,000

(i) Your net income tax is 0

(ii) Your net income tax is 2500

Q5 Write a program to determine whether a character entered by the user is lowercase or not

Code: #include <stdio.h>

Ent main () {

if 97-122 = a-z

char ch;

printf ("Enter the character \n");

scanf ("%c", &ch);

if (ch <= 122 && ch >= 97) {

printf ("It is lowercase");

else {

printf ("It is not lowercase");

return 0;

Ans: Enter the character



(i) A

(ii) a

(iii) @

- (i) gt es not lower case
- (ii) gt qs lower case
- (iii) gt qs not lowercase