C Instructions

Discussion

- What is an instruction in real life?
- Why do you need instructions in real life?
- Different types of instructions you come across?

Three types of instructions

- Type declaration instruction
- Arithmetic instruction
- Control instruction

Type Declaration

- To declare variables of any type
- All variables must be declared before using them
- Eg:

```
int a,b;
float j=9.0;
char name='R';
```

Arithmetic Instruction

- To perform arithmetic operations
- variable on left of = and variable and/or constants on the right
- Operations: +,-,%,/,*
- No operation for ^
- Type conversion
- Priority - * / % , + - ,=
- Associativity – L to R , R to L

Control Instructions

- Sequence control
- Selection or Decision control
- Repetition or Loop control
- Case control

Decision control

- if
- if else
- nested if else
- else if

- equality operator: ==
- inequality operator: !=
- logical operators : &&, ||, !
- Conditional operator: exp1?exp2:exp3

Precedence

- (), [], ., ->,++,-- (Postfix) (LR)
- ++,-- (Prefix),+,-(Unary),!,~,(typecast),*,& (RL)
- /,%, * (LR)
- +,- (Arithmetic) (LR)
- <<,>> (LR)
- <,<=,>,>= (LR)
- ==,!= (LR)
- &,^,|,&&,|| (LR)
- ?: (RL)
- All assignments (RL)

Associativity:

Right to Left (RL)

Left to right (LR)

Loops

- while
- do while
- for

Jump statements:

- break
- continue

Case Control

- switch

Important points:

- break
- continue
- case keyword
- default case
- no condition only constant

Summary

- There are three ways for taking decisions in a program. First way is to use the **if**-**else** statement, second way is to use the conditional operators and third way is to use the **switch** statement.
- The default scope of the **if** statement is only the next statement. So, to execute more than one statement they must be written in a pair of braces.
- An **if** block need not always be associated with an **else** block. However, an **else** block is always associated with an **if s**tatement.
- If the outcome of an **if-else** ladder is only one of two answers then the ladder should be replaced either with an **else-if** clause or by logical operators.
- **&&** and **||** are binary operators, whereas, **!** is a unary operator.
- In C every test expression is evaluated in terms of zero and non-zero values. A zero value is considered to be false and a non-zero value is considered to be true.
- Assignment statements used with conditional operators must be enclosed within a pair of parenthesis.

Summary

- The three type of loops available in C are **for**, **while**, and **do-while**.
- A **break** statement takes the execution control out of the loop.
- A **continue** statement skips the execution of the statements after it and takes the control to the beginning of the loop.
- A **do-while** loop is used to ensure that the statements within the loop are executed at least once.
- The **++** operator increments the operand by 1, whereas, the -- operator decrements it by 1.
- The operators +=, -=, *=, /=, %= are compound assignment operators. They modify the value of the operand to the left of them.

Summary

- When we need to choose one among number of alternatives, a **switch** statement is used.
- The **switch** keyword is followed by an integer or an expression that evaluates to an integer.
- The **case** keyword is followed by an integer or a character constant.
- The control falls through all the cases unless the **break** statement is given.
- The usage of the **goto** keyword should be avoided as it usually violets the normal flow of execution.

Question

output:?

```
Int i=4;
Switch(i)
{
        default: printf(“Hi\n”);
        case 1: printf(“all\n”); break;
        case 2: printf(“all2\n”); break;
        }
```

- Point out the error?

```
Int I =0;
For (;;)
{
        printf(“%d”;i++);
        If (i>10) break;
}
```

Imp Question

- What is the output

```
Int main()
{
char i=0;
for (i<=5 && i>=-1; ++i; i>0)
        printf (“%d”,i);
return 0;
}
```

Imp

- Output?

Int i=5;

While (i-- >=0) printf(“%d”,i);

- Output?

Float a = 0.7

If(0.7>a) printf(“Hello”);

Else printf(“Get lost”);

- Output?

Int i=7;

Float j=7.0;

if(i==j) printf(“Hello”);

- Output?

Int i=10,j=15;

If (i%10=j%15) printf(“Equal”);

Else printf(“Get lost”)

- Output?

a=0,b=1,c=3

*((a)?&b:&a)=a?b:c;

printf(“%d %d %d \n”, a, b, c)

- Output

char j=1;

while (j<=255)

{printf(“%d”,j);j=j+1;}

- Output

```
int x=1,y=1;
for( ; y ; printf("%d %d\n", x, y))
        y = x++<=5
```

- Output

```
Int i=-3,j=2,k=0,m;
m=++i || ++j && ++k;
printf( "%d %d %d %d" , i, j, k, m );
```

finalDesk

Answers

- Hi

  All
- No error
- 1 2 3….126 127 -128 -127 . … -2 -1
- 4 3 2 1 0 -1
- Hello (Recurring binary storage of 32 and 64 bits)
- Hello (Automatic type promotion)
- Error: Lvalue required

- 3 1 3
- Infinite loop
- 21 31 41 51 61 70
- -2 2 0 1

Contact Info

- trainers@finaldesk.com
- rishabh@finaldesk.com
- nilesh@finaldesk.com
- jignesh@finaldesk.com
- yash@finaldesk.com
- anand@finaldesk.com