if/else statements:

syntax:

```
if ( condition )
{ statement; }
```

These brackets are not necessary for single line statement but for multiline we require them.

```
else
{statement;}
```

Day: Date: / /

- if the condition is true then the statements under if are run else they are not run and statements under else are run.

- An if can be without an else but ~~else~~ else cannot be without if.

examples

1)
```
if (a > b)
    cout << " a>b " << endl;
    cout << " a is greater " << endl;
else
    cout << " b > a " << endl;
```

Output:

error . ~~else~~ else is without an if.

How?

we did not add curly brackets ~~after~~ for the if statements and so the first line cout << a>b -- is inside if. The second line cout << "a is greatr" is independent which leaves else with an if as else should come right after if.

```
if (a>b)
    {cout << " a>b " << endl;}
cout << " a is greater " << endL;
else
    cout << " b > a " << endl;
```

This is in if only

This becomes independent

else without an if.

Yousaf

Day: Date: / /

2)
```
if (a >b)
   if (a>c)
      cout<< " a is greatest" <<endl;
else
   cout<<" b greater than a"<<endl;
```

Now this is a logical error. This will be compiled. ~~but~~

we wanted the else to be with the first if but else is always with the nearest if.

Correction:

```
if (a>b)
  {if (a >c)
     cout << " a is greatest "<< endl;}
else
  cout<<" b is greater than a"<<endl;
```

IMPORTANT POINTS

- If in question logical operators not allowed then we can make and ~~log~~ (&&) by nesting if's in true. ie. if() because this is what and does.
```
if()
  if()
  { }
```

-- for OR it is in else statement

Yousuf

Day: Date: / /

if-else if:

- Instead of nesting if statements we can use this approach for a better readibility and easier understanding.

```
if (condition(s))
{
    statement 1;
    statement 2;
}
else if (condition(s))
{
    statement 1;
    statement 2;
}
else if (condition(s))
{
    statement 1;
    statement 2;
}
else
{
    statement 1;
    statement 2;
}
```

if this condition true then no else if is run.

if this condition true then only its statements shown.

same as above

if nothing true, this is run.

else is optional.

Yousaf

Day: Date: / /

example:
Write a code to show Mon, Tue or Wed when a user enters 1, 2 or 3.

```
int main()
{
   int num =0;
   cout << " enter a number: " << endl;
   cin >> num;

  if (num== 1)
    { cout << " Monday " << endl;}
  else if (num== 2)
    { cout << " Tuesday " << end l;}
  else if ( num == 3)
    {cout << " Wednesday " << endl;}
else
   { cout << " Not a valid number " << endl;}

  return 0;
}
```

- else if statement work as OR.

Yousaf

Day: Date: / /

Ternary operator / conditional operator:

condition ? True : False

if condition is true then this statement is run

if condition is false then this statement is run

example:

1) a > b ? cout << "a is greater" << endl; : cout << "a is smaller" << endl;

We can also nest it

2) a > b ? a > c ? cout << "a" << endl; : b > c ? cout << "b"; : cout << "c";

cout << "c"; :

IMPORTANT POINTS

* True and False both should be of same datatype. (both should be int or both should be string).
* It should only be used for simple if/else single line statements.

Yousaf

Date: / /

Switch ; case:

- This is also an alternative to if-else.
- It has only one input
- This works best for creating Menus.

example:
Write a program to ~~show~~ give user ~~options of types~~ details of what type of ~~things~~ car they select.

```
int main ()
{
   int type = 0;
   cout << "Which model do like:\n
          1. 124\n
          2. D31\n
          3. 322" << endl;
   cin >> type;

   switch ( type )        <- one input only.
   {
      case 1 : cout << "a beautiful blue car, V8 engine" << endl;
               break;
      case 2 : cout << "red car, V6 engine" << endl;
               break;
      case 3 : cout << "white car, V12 engine" << endl;
               break;
      Default: cout << "invalid number" << endl.
   }
      return 0;
   }
```

Yousuf

Date: / /

- Switch works by equality only e.g
  if type entered was 1
  then 1==1 case is checked.

- When no case is equal then default answer is given so default works like an if.

Break:

- Break is needed after each under every case mostly.

if break not added then whichever case gets true, all ~~cout~~ statements are printed even of other cases and also default till a break is reached.

example:

In car example if we didn't add break and type was entered as 2 then output.

red color, V6 engine
White color, V12 engine
Invalid number.

So break is important.

Yousaf

Date: / /

How to use inequalities for small intervals.

example. if BMI greater than 30 but less than 35 we then output "eat less, exercise".
if BMI greater than = 35 then "Medical advice"

```
Switch (BMI)
{
    case 30:
    case 31:
    case 32:
    case 33:
    case 34: cout "eat less, exercise" << endl;
             break;
    default: cout "Seek Medical advice"
}
```

Now if any case from 30-34 becomes true then then all statements become true till break.
If type is 35 or greater then default option is used.

Date: / /

Some points for conditional:

1) if we have some assignment in if condition:

if (n = 1) → True

if (n = 0) → False

if we assign 0 to any value in the condition the condition will become false.

Think of it like, firstly the value is being assigned and then the value of n is being checked.

* First Value is assigned.
* Then The condition is checked.

so if.

```
1  int n = 2
2  if (n = 0) {
3      int n = 3;
4      cout << ++n << endl;
5  }
6  cout << n;
```

Output:

4

if(n=0) even if condition is false n=0 is n's value.

We can declare variable again inside another block. so ++n gives 3+1 = 4.

2 n was assigned 0 in the condition so statements the block of if n = 0.

Date: / /

How to use inequalities for small intervals.

example: if a BMI greater than 30 but less than 35 we then output "eat less, exercise".
if BMI greater than = 35 then "Medical advice".

```
switch (BMI)
{
    case 30:
    case 31:
    case 32:
    case 33:
    case 34: cout "eat less, exercise" << endl;
             break;
    default: cout "Seek Medical advice!
}
```

Now if any case from 30-34 becomes true then all statement become true till break.
If type is 35 or greater then default option is used.

Date: / /

Some points for conditional:

1) if we have some assignment in if condition:

if (n = 1) → True

if (n = 0) → False

if we assign 0 to any value in the condition the condition will become false

Think of it like, firstly the value is being assigned and then the value of n is being checked

* First Value is assigned.
* Then The condition is checked.

so if.

```
1 | int n = 2
2 | if (n = 0) {
3 |     int n = 3;
4 |     cout << ++n << endl;
5 | }
6 | cout << n;
```

if(n=0) even if condition is false n=0 is n's value.

Output:

4 we can declare variable again inside another block. so ++n gives 3+1 = 4.

0 as n was assigned 0 in the condition so obviously the block of if n = 0.

Yousaf

Day:

Date: / /

2) Ternary:

* If we are using cout in true or false then both true or false should contain couts.
  Cout can be in any way.
  e.g

  condition ? cout << "Hi" : cout << (a+b);

  allowed.

* We can also use cin but both statements should be cin.

We cannot even use cin and cout together; one is in true and other is in false.