Selections :

Selections are primarily driven by means of conditions. Conditions are evaluated Boolean statements that can be a single comparison or complex Boolean statement with properties including any number of combinations of the AND, OR, and NOT.

One Way :

if(condition){ /\* code \*/ }

the simplest form of the selections in the one way selection which we use the if statement to represent. The condition is checked to be seen if true or not, and then the code that exists inside the braces under the if is run. The condition is true, the code is run, if not, it is ignored.

Two Way :

if(condition){ /\* code \*/}

else{ /\* code 2 \* / }

the first type of difference is the two way selection, we use this to represent the option of one or the other. If the condition is not true, then code 2 is run, if the condition is true, code is run. Simple as that, one segment of code is ignored while the other runs.

Multiple way :

if – else if – else

switch

These are types of statements that on higher level programs are more common, due to the options being more diverse and important at moment of comparisons. You need to not only do something if something is true, but if something else is true you need a different option for each time there is a difference and you don’t want to re-evaluate the first conditions.

Nesting conditions :

When nesting conditions you are making a statement that is equivalent to an and with the outside condition.