Control Statements:

**• if / else :**

“if statement” is used to perform conditional checks.

**a.**

if (condition)

{

Statement 1

}

Statement X;

In the 1st scenario first condition will be checked. If the condition is true the statement 1 will be executed and then control will be transferred to statement X. If condition is false then control will be transferred to statement X without executing statement **1**.

**b.**

if (condition)

{

Statement 1

}

else

{

Statement 2

}

Statement X;

In the 2nd scenario if the condition is true then statement 1 will be executed and the control will be transferred to statement X. If the condition is false then statement 2 will be executed and control will be transferred to statement X. Here either if or else condition will execute and both of them will never execute together.

**c.**

if (condition)

{

Statement 1

}

else if (condition)

{

Statement 2

}

else

{

Statement 3

}

Statement X;

In the 3rd scenario if the condition is true then statement 1 will be executed and then control will be transferred to statement X. If the condition is false it will come to else if part and checks the condition. If true then Statement 2 will be executed and control will be transferred to statement X. If false then statement 3 will be executed and control will be transferred to statement X.

Ex: //Defining & initializing personAge variable to 20

int personAge = 20;

//Checking weather personAge variable contains less than 10 value

if(personAge < 10)

{

NSLog(@"It's a kids Age");

}

//Checking weather personAge variable contains less than or equal to 20 value

else if(personAge <= 20)

{

NSLog(@"It's students age");

}

//Checking weather personAge variable contains greather than or equal to 25 value

else if(personAge >= 25)

{

NSLog(@"It's Workers Age");

}

//If personAge variable's value not satisfyies the above conditions

//It prints this message

else

{

NSLog(@"There is nothing special about this age");

}

**• for**

• for statement is used to execute a set of statements multiple times.

1 2 4

for (initialization;condition;increment/decrement)

{ 3

Statement 1;

}

• It is alternative to while loop. In while loop initialization, condition, statements and increment/decrement are not defined in single statement.

• When for statement is encountered 1st initialization statement will be executed and then condition will be verified. If condition is true then statements inside for block will be executed and then increment or decrement operation will be executed. After increment or decrement operation again condition will be verified. If the condition is true then again for block will be executed and same process will be repeated till condition becomes false. If the condition is false then control will come out the loop and executes statement X

**• switch / case :**

• Switch statement is used to execute a particular set of statements among multiple conditions.

• It is alternative to complicated if else if ladder conditions.

• The expression type in the switch must be any of the following data type.

• break is optional. Every case must end with break statement which will help to terminate and transfer the control outside of switch block. If no break is used then execution will continue to next case.

• default is optional. If present it will execute only if the value of the expression does not match with any of the case and then control goes to statement X. If not present then control will exit switch statement and goes to statement X.

**• while and do/while :**

**while:**

• while is an entry controlled loop. In while statement 1st condition will be verified. If the condition is true loop will be repeated .If the condition is false then control will come out of the loop.

**do/while:**

• do/while is exit controlled loop. In do while first all the statements inside the block will be executed and then condition will be verified. If the condition is true then loop will be repeated. If condition is false then control will come out of the loop and execute statement X.

• In while when the condition is false for the first time then the statement inside the block will be executed for 0 times, whereas in do while statement will be surely executed for one time.

**• break :**

• It helps to transfer control to another part of the program.

• It can be used in switch statement or in do, while and for loops. It also can be used in labeled blocks.

• It has the following uses.

• It helps in terminating a switch statement.

• It is used to exit a loop or force immediate termination of a loop bypassing the conditional expression and other remaining code in the loop.

**• continue :**

• It is used to move the control to the beginning of the loop.

• When continue is used in while or do-while loop then control is directly transferred to the test condition that controls the loop.

• When continue is used in for loop control is transferred to iteration portion first then to the test condition.

**Note:** break is used to move the control to the end of the loop but continue is used to move the control to the beginning of the loop.

**• return :**

• return keyword terminates the execution in a method and returns the control to the caller method.