

Control	Pseudo code	Flow diagram	C code
if selection	if condition is met then perform statements	<pre> graph TD     Start(( )) --&gt; Condition{condition}     Condition -- "condition zero (false)" --&gt; Merge(( ))     Condition -- "condition nonzero (true)" --&gt; TrueStatements[true statements]     TrueStatements --&gt; Merge     Merge --&gt; Exit(( )) </pre>	<pre> <b>if</b> (condition)     true statement; </pre> <p>OR</p> <pre> <b>if</b> (condition) {     true statements; } </pre>
if ... else selection	if condition is met then perform statements otherwise perform statements	<pre> graph TD     Start(( )) --&gt; Condition{condition}     Condition -- "condition zero (false)" --&gt; FalseStatements[false statements]     Condition -- "condition nonzero (true)" --&gt; TrueStatements[true statements]     FalseStatements --&gt; Merge(( ))     TrueStatements --&gt; Merge     Merge --&gt; Exit(( )) </pre>	<pre> <b>if</b> (condition)     true statements; <b>else</b>     false statements; </pre> <p>OR</p> <pre> <b>if</b> (condition) {     true statements; } <b>else</b> {     false statements; } </pre>
while repetition	while condition is met repeatedly perform actions	<pre> graph TD     Start(( )) --&gt; Condition{condition}     Condition -- "condition zero (false)" --&gt; Exit(( ))     Condition -- "condition nonzero (true)" --&gt; RepeatedStatements[repeated statements]     RepeatedStatements --&gt; Condition </pre>	<pre> <b>while</b> (condition)     repeated statement; </pre> <p>OR</p> <pre> <b>while</b> (condition) {     repeated statements; } </pre>

do while repetition	repeatedly perform actions while condition is met	<pre> graph TD     Start(( )) --&gt; Statements[repeated statements]     Statements --&gt; Condition{condition}     Condition -- "condition nonzero (true)" --&gt; Statements     Condition -- "condition zero (false)" --&gt; End(( )) </pre>	<pre> do     repeated statement; while (condition); </pre> <p>OR</p> <pre> do {     repeated statements; } while (condition); </pre>
for repetition	initialize evaluate condition if condition repeatedly perform actions perform increment re-evaluate condition	<pre> graph TD     Start(( )) --&gt; Init[initialize]     Init --&gt; Cond{continuation condition}     Cond -- "condition nonzero (true)" --&gt; Statements[repeated statements]     Statements --&gt; Inc[Increment(s)]     Inc --&gt; Cond     Cond -- "condition zero (false)" --&gt; End(( )) </pre>	<pre> for(initialize;     condition;     increment) statement; </pre> <p>OR</p> <pre> for(initialize;     condition;     increment) {     statements; } </pre>
switch selection	for each case, from the first case, evaluate a value against cases. if the value equal to case perform actions if "break", exit switch else assume all remaining cases true else evaluate next case. if "default" reached perform default actions exit switch	<pre> graph TD     Start(( )) --&gt; Case1{case 1: (val == 1?)}     Case1 -- "condition nonzero (true)" --&gt; True1[true statements]     True1 --&gt; Break1[break]     Case1 -- "condition zero (false)" --&gt; Case2{case 2: (val == 2?)}     Case2 -- "condition nonzero (true)" --&gt; True2[true statements]     True2 --&gt; Break2[break]     Case2 -- "condition zero (false)" --&gt; Default[default statements]     Break1 --&gt; Join(( ))     Break2 --&gt; Join     Default --&gt; Join     Join --&gt; End(( )) </pre>	<pre> switch ( val ){     case 1 :         actions(s);         break;     case 2 :         actions(s);         break;     default:         actions(s); } </pre> <p>Note:  1. If the <b>break</b> is omitted in a case, the subsequent cases will all evaluate and do so as if true.  2. All the cases need to be constant values.  3. default is optional</p>