exit()
if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

Conditional execution / decision branching

Comp Sci 1570 Introduction to C++

Outline

exit()

if else
- Simple if
- if else
- Nested if else
- else-if ladder
- Points to remember
- Examples

1. exit()

2. if else
   - Simple if
   - if else
   - Nested if else
   - else-if ladder
   - Points to remember
   - Examples

exit()
if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

The exit() Function

- In main, when the return 0; statement is executed, the value 0 is sent to the operating system, signaling the successful completion of the program.
- Other values that might be sent to the OS can signal other conditions, such as an unsuccessful completion, and particular error conditions.
- Normally, there should be only one return statement in the main function.
- When exit() is executed, all the necessary "cleaning up" after your program is done and the value you designate to be sent by exit is sent to the OS.
- The statement exit(0); sends the int value 0 back to the OS.
- Different integer values can be used to mean different error conditions.
- We will see a good use for the exit function with loops

Outline

exit()

if else
- Simple if
- if else
- Nested if else
- else-if ladder
- Points to remember
- Examples

1 exit()

2 if else
- Simple if
- if else
- Nested if else
- else-if ladder
- Points to remember
- Examples

Outline

exit()

if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

1 exit()

2 if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

if

exit()

if else

Simple if

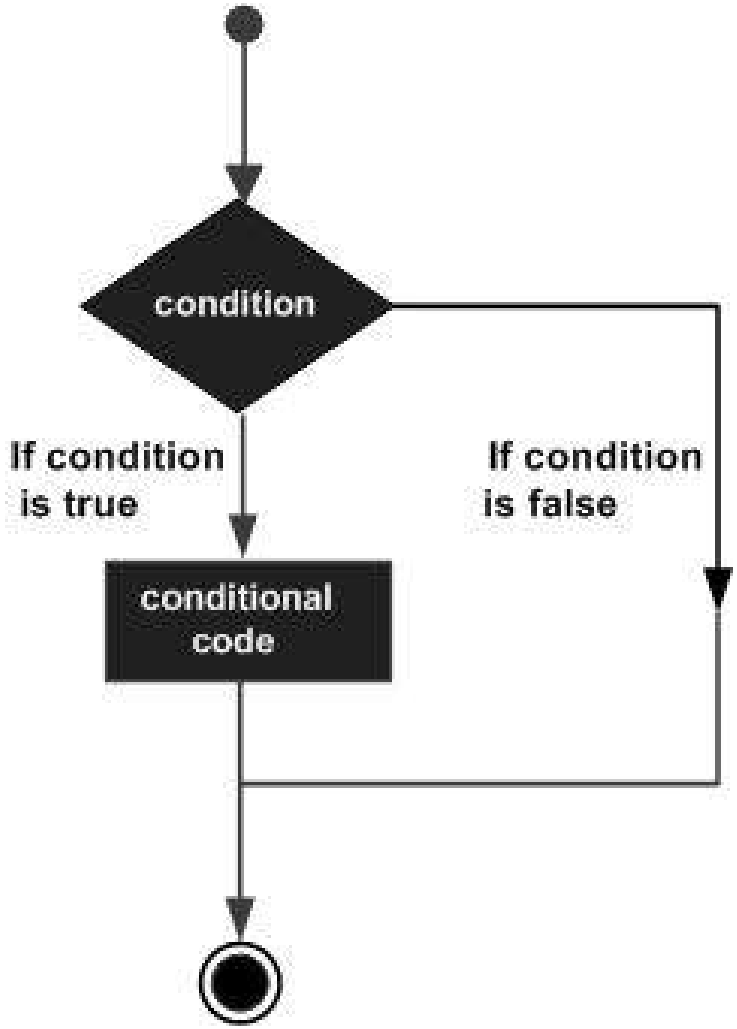
if else

Nested if else

else-if ladder

Points to remember

Examples



Simple if statement

exit()
if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

```
if( expression )
{
   statement-inside;
}
statement-outside;
```

- Expression is called a conditional expression, which is any syntactically correct C++ expression that evaluates to true or false or a number
- If the expression evaluates to true (non-zero), then 'statement-inside' it will be executed, otherwise 'statement-inside' is skipped and only 'statement-outside' is executed.
- if and else are reserved words and begin with lower case.
- statements are either simple (single) C++ statements (with the ; included), or a compound statement (with all semi-colons included). A compound statement is a sequence of statements all of which are to be executed in order, and delineated by { }

Outline

exit()

if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

1 exit()

2 if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

if...else statement

exit()
if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

```
if( expression )
{
 statement-block1;
}
else
{
 statement-block2;
}
```

- If the 'expression' is true, the 'statement-block1' is executed, else 'statement-block1' is skipped and 'statement-block2' is executed.
- If the expression evaluates to true (non-zero), If the expression evaluates to false, the else statement is executed if it exists.

Outline

exit()

if else
- Simple if
- if else
- Nested if else
- else-if ladder
- Points to remember
- Examples

1 exit()

2 if else
- Simple if
- if else
- Nested if else
- else-if ladder
- Points to remember
- Examples

Nested if...else

exit()
if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

The general form of a nested if...else statement is,

```
if( expression )
{
   if( expression1 )
   {
      statement-block1;
   }
   else
   {
      statement-block2;
   }
}
else
{
 statement-block3;
}
```

- if 'expression' is false the 'statement-block3' will be executed, otherwise it continues to perform the test for 'expression 1' .
- If the 'expression 1' is true the 'statement-block1' is executed otherwise 'statement-block2' is executed.

Outline

exit()

if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

1 exit()

2 if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

exit()
if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

else-if ladder

The general form of else-if ladder is,

```
if(expression 1)
{
 statement-block1;
}
else if(expression 2)
{
 statement-block2;
}
else if(expression 3 )
{
 statement-block3;
}
else
 default-statement;
```

- The expression is tested from the top(of the ladder) downwards.
- As soon as the true condition is found, the statement associated with it is executed.

Outline

exit()

if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

1 exit()

2 if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

Points to Remember

exit()
if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

- No curly braces are required in the above case, but if we have more than one statement inside if condition, then we must enclose them inside curly braces.
- == (equality) must be used for comparison in the expression of if condition, if you use = (assignment) the expression will always return true, because it performs assignment not comparison. Other than 0(zero), all other values are considered as true.

Outline

exit()

if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

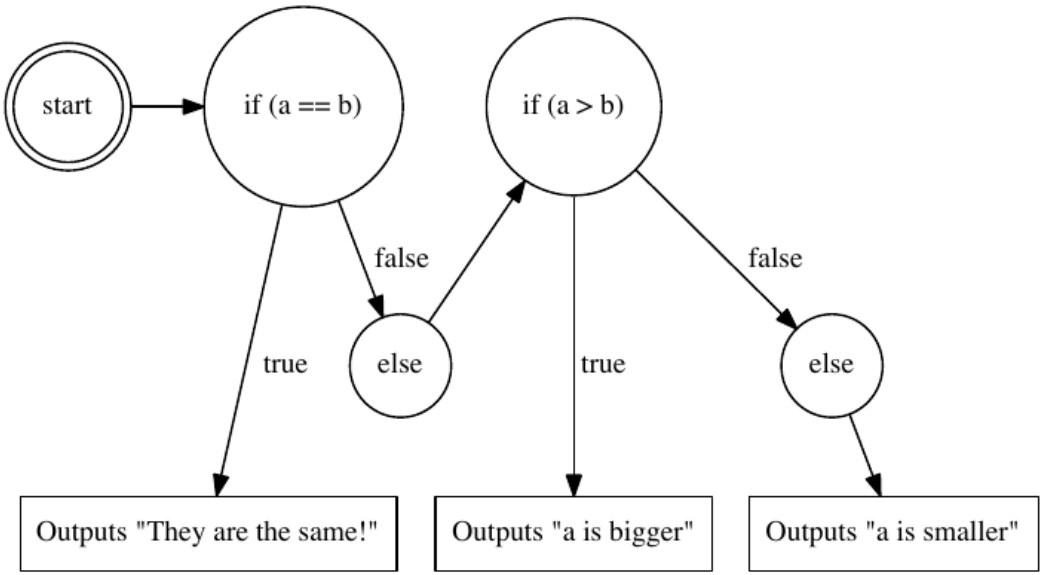
1 exit()

2 if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

Flowchart logic

exit()
if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples

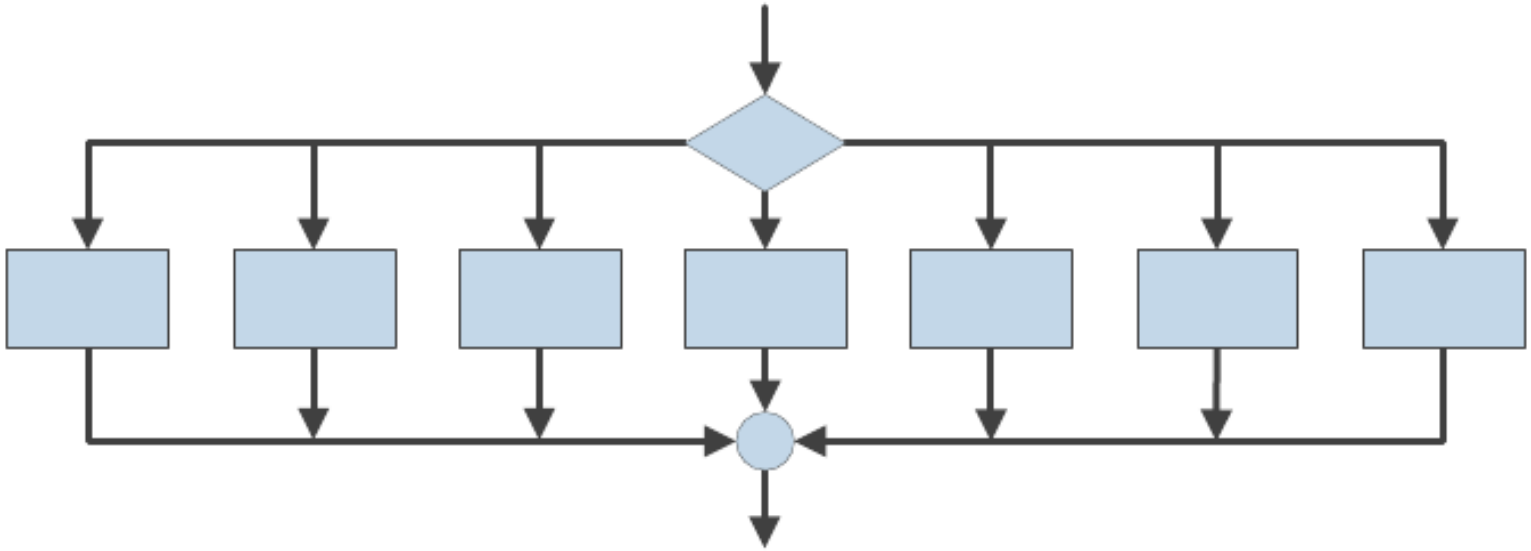
```
if(a == b)
{
  cout << "They are the same!" << endl;
}
else if(a > b)
{
  cout << "a is bigger!" << endl;
}
else
{
  cout << "a is smaller!" << endl;
}
```

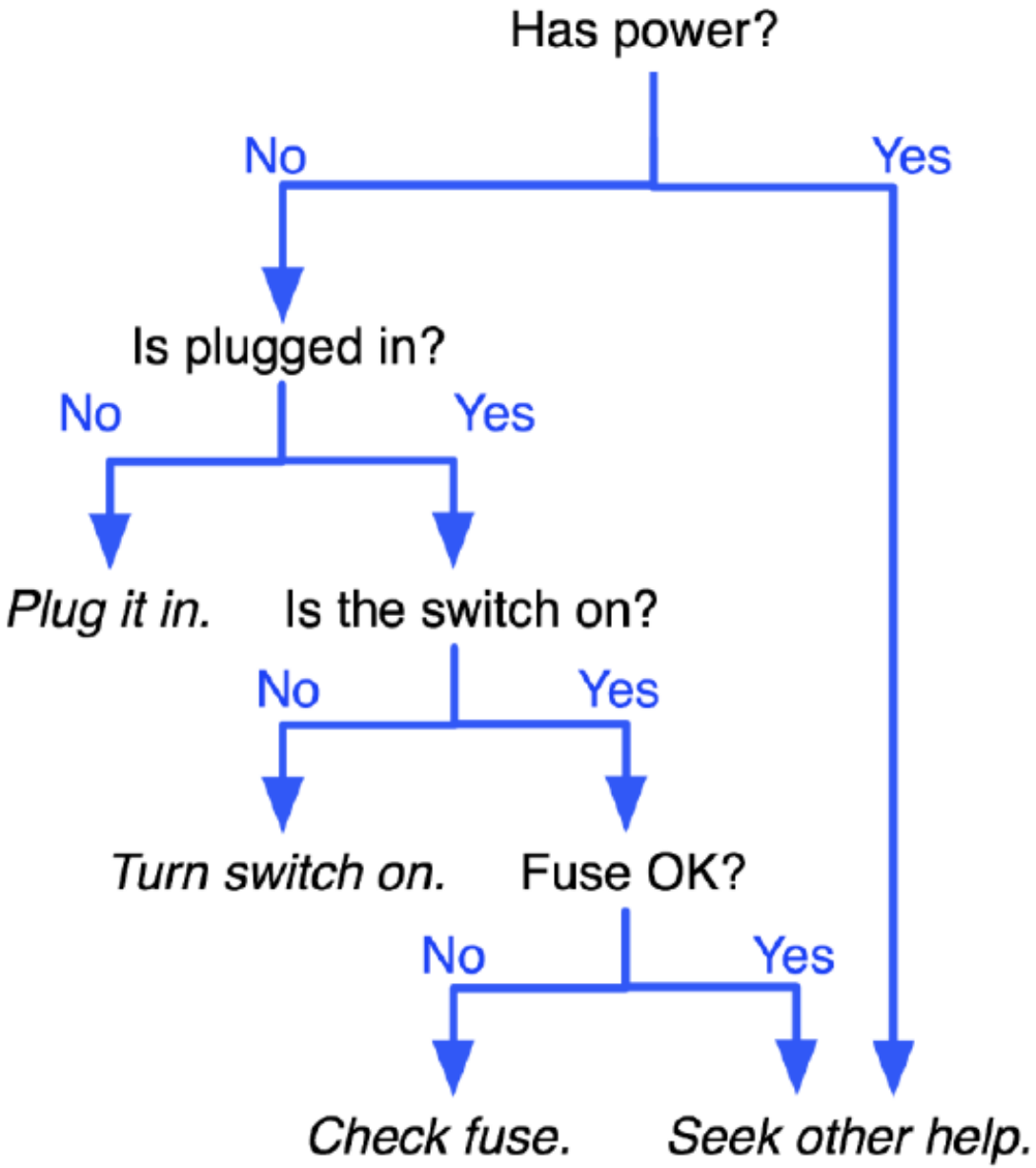


Flowchart logic

exit()

if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples



Flowchart logic

exit()
if else
Simple if
if else
Nested if else
else-if ladder
Points to remember
Examples