PH502: Scientific Programming Concepts

Irish Centre for High End Computing (ICHEC)

September 23, 2020

Overview

- This lecture we will cover conditional statements.
- In the previous lecture we used a conditional statement to terminate a loop.
- In general these statements allow the programmer to control which parts of the code are run.
- Logical expressions are an essential part of these statements.

Conditional Statements

- Conditional statements allow different blocks of code to be executed given a logical variable or expression is true or false.
- When a query can be answered with a Yes, we say that the statement contained in the query is True. If we answer the query with a No, we say that the statement contained in the query is False.
- We will introduce the
  1. if-then,
  2. switch-case structure.
- **if-then**: allows us to check whether a statement is true or not.
- **switch-case**: permits us to execute different statements based on the different values of a parameter.

If Statements

- If a given statement is true we want to execute a particular command. However, if it is not true we do not want to do anything.

```
if (statement) {
   commands to be executed
}
```

```
if (statement) then
  commands to be executed
endif
```

- The second situation is when we want to execute one set of commands when the statement is true, and another set of commands when it is not.

```
if (statement) {
   commands to be executed
}
else {
   commands to be executed
}
```

```
if (statement) then
  commands to be executed
else
  commands to be executed
endif
```

If Statements

- An example of this is using an "if else" statement.

```
if (x > 1.0) {
    z = y/x;
}
else {
    z = 0.0;
}
```

```
if (x .gt. 1.0) then
  z = y/x
else
  z = 0.0
endif
```

- These statements can be combined to give,

```
if (i == 0) {
    z = x + y;
}
else if (i == 1) {
    z = x - y;
}
else {
    z = x * y;
}
```

```
if (i .eq. 0) then
  z = x + y
else if (i .eq. 1) then
  z = x - y
else
  z = x * y
endif
```

- conditional operator: Exp1 ? Exp2 : Exp3;

Comparing Floating Point Numbers

- When comparing floating point numbers contained in two locations, all comparisons are on the real number line in terms of their position.
- When comparing numbers the statements can be of the following forms: $x == y, x! = y, x > y, x >= y, x < y, x <= y$.
- When comparing numbers, representation and arithmetic errors should be considered.

```
float x,y;
// Not good
if (x == y) {
  do something}

// Better
if (fabs(x-y) < 0.001) {
  do something }
```

```
real (kind=4) :: x,y
if (x .eq. y) then
  do something
endif

if (abs(x-y) .lt. 0.001) &
                        then
  do something
endif
```