Selection: The IF Statement

|  |  |
| --- | --- |
| C# Instructions covered previously covered:   * **Variables:** <data type> <name>; * Console.WriteLine(“ “); * Console.ReadLine(); * **Operations:** +, -, \*, / | New C# Instructions used in these exercises:   * If <condition> * else |

# An example If Statement

|  |  |
| --- | --- |
|  | **Annotation Area** |

## The IF Statement

So far the programs that you have written have followed set instructions. Selection allows you to write programs with choices for the user. Programs can also be written that display different options/results depending on the data that has been entered.

All if statements have a **condition** that must be **true** in order for the **code to run**

For example:

**IF** Number is less than 0 **THEN IF** (Number <0) **THEN**

{

Number is negative console.writeLine(“Number is negative”);

}

**ELSE ELSE**

{

Number if positive console.writeLine(“Number is positive”);

}

## Operators

The operators that you can use for the **condition** part of an if statement are shown in the table below

|  |  |
| --- | --- |
| == | Equal to |
| != | Not equal to |
| < | Less than |
| <= | Less than or equal to |
| > | Greater than |
| >= | Greater than or equal to |

# Else If

If statements can be extended to give more possibilities using ‘else if’.

**In the previous example what would happen if ‘0’ was entered?**

The example code below gives an appropriate output if ‘0’ is entered

**IF** Number is less than 0 **THEN IF** (Number <0) **THEN**

**{**

Number is negative console.writeLine(“Number is negative”);

**}**

**ElSE IF** Number is 0 **THEN ELSE IF** (Number = 0) **Then**

**{**

Number is Zero console.writeLine(“Number is zero”);

**}**

**ELSE ELSE**

**{**

Number if positive console.writeLine(“Number is positive”);

**}**

## Scenario

When you try to take money out of an ATM (Automatic Teller Machine, commonly called a cash dispenser or ‘hole in the wall’), you are only allowed to withdraw cash up to your credit limit.

For example, if your credit limit is £100, and you try to withdraw £50, then it should work fine. However, if you try to withdraw £150, you will not be allowed to, and a message will appear on the screen advising you that this is over your credit limit.

## Task1: Selection

In the space below write an if statement for the ATM scenario

|  |  |
| --- | --- |
| Pseudo Code | C# Code |
| If number is more than 100 THEN  Display withdrawal is over limit  ELSE  Withdraw | {  int withdraw;  Console.WriteLine("type how much money you would like to withdraw in numbers (WITHOUT THE POUND SIGN)");  withdraw = int.Parse(Console.ReadLine());  if (withdraw > 100)  {  Console.WriteLine("you have gone over the withdrawal limit of 100 please input a number less than 100");  }  else  { Console.WriteLine("You have Withdrawn £" + withdraw + " from your account.");  }  }  }  } |

## Task2: Design

You are going to design, implement and test a program for the display of an ATM/Cash Machine

### Hierarchy Charts

So far we have used Pseudo code to plan programs. Hierarchy Charts can also be used to plan/design programs in a visual way. Hierarchy Charts are read top down and line by line from left to right. The Hierarchy Chart for the ATM program is shown below:

Program: ATM

Ask user how much money they would like to withdraw

Display ‘thank you for using ATM’

Check withdrawal amount

Get withdrawal amount

Withdraw amount <= 100 withdraw amount >100

Display please take your cash

Display exceeded daily limit message

The way a hierarchy charts represents a **selection** process is:

Check condition

Condition is true Condition is false

Perform Action A

Perform Action B

**Write the Pseudo Code below for the ATM scenario. Variables have been started for you**

**Variables:**

CashRequest : double/integer

**Pseudo Code:**

|  |
| --- |
|  |

## Task3: Implementation

Use the Pseudo Code and the Hierarchy Chart to help you complete the ATM program in Delphi.

This time you do not have a starting point.

Look back at your notes and previous programs you have written if you need help starting.

## Task 4: Testing

Use the table below to test your program. You also need to choose 3 of your own tests.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected Output** | **Actual Output** | **Comment** |
| **Cash** | **Message** | **Message** |  |
| **Normal data** | 50 | Please take you money |  |  |
|  |  |  |  |
| **Boundary data** | 99 | Insufficient funds available |  |  |
|  |  |  |  |
| **Erroneous data** | -50 | Incorrect value |  |  |
|  |  |  |  |

## Extension Tasks

1. You have been commissioned to design and implement a program that will select a suitable prize, depending on which number between 1 and 5 is entered by the user. If the number 1 is entered the price should be a TV and if the number 3 is entered the price should be £1000. Any other number should result in a message explaining that the user has not won this time.

|  |
| --- |
| Hierarchy Chart |
| Pseudo Code |

1. Create a calculator that gives the user the option to add, subtract, multiply or divide 2 numbers together. Choose either a Hierarchy Chart or Pseudo Code to plan your solution

|  |
| --- |
|  |

# Self-Study Tasks (Programming 2)

Tick each box once you have completed a task to help keep organised

* Read class notes/hand out to make sure that you fully understand the content
* Read/complete the following sections of the online tutorial: 
  + The If Statement

**Programming Tutorial can be found on the vle, the shared area or by visiting** [**http://csharp.net-tutorials.com/basics/introduction/**](http://csharp.net-tutorials.com/basics/introduction/)

* Read/complete Seletion if-else online tutorial

**This can be found on the vle, the shared area or by visiting** [**https://msdn.microsoft.com/en-us/library/5011f09h.aspx**](https://msdn.microsoft.com/en-us/library/5011f09h.aspx)