Task 2 - The Report

In this report I will explain sequence selection and iteration as used in programming, as well as the factors that influence the choice of programming languages.

Part 1 - Explaining a Sequence Statement

A sequence statement is anything that happens in a sequence. Mostly executed inside other statements.

```Javascript

var variable = 10; document.getElementById.innerHTML = variable; var x = variable + 10; ```

As you can see, These statements are all on the same line, and are simple. They only execute once.

Here is an example of sequence in C#. (Just a side note, asking people to screenshot code is useless and awkward for everyone.)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            double num;
            if (textBox1.Text != "" && textBox2.Text != "" && double.TryParse(textBox2.Text, out num) && double.TryParse(textBox1.Text, out num))
            {
                long num1 = (long)Convert.ToDouble(textBox1.Text);
                long num2 = (long)Convert.ToDouble(textBox2.Text);
                long answer = num1 + num2;
                label1.Text = answer.ToString();
            }
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }
    }
}
```

---

## Part 2 - Explaining a Select Statement

A select statement is a statement that selects a value for comparison. Examples of select statements include: if, else if, else, and while.

Here is an example of some code that checks is a number is validity by the system's terms:
```

```JavaScript

function isNumber(n) {
  return !isNaN(parseFloat(n)) && isFinite(n);
}
function validateNum(num){ //This defines a function for validating a number.
  if (num == "") {// Checks that the number has been entered (1)
    return false;
  }
  else if (!isNumber(num)) { // Checks if the number is 6 characters long( 2)
    return false;
  }
  else if (!num >= 0){
    return true;
  }
  else { // ( 3)
    return  true;// if everything checks out, the function returns true
  }
}
function myFunction() {// this function is called by a button in HTML
  var message;
  var num = prompt("Please enter a number","");// Asks the user for input
  if (validateNum(num)){
    message = "Hello. The number "+ num +"is a valid number." ;
  }
  else{
    message = "Please click again and enter a valid number";
  }
  $("#greeting").html(message);// Publishes the message to an HTML element
}

```

If you look find the line with '(1)' at the end, you will see the first if statement. Javascript if (num == "") {// Checks that the number has been entered (1) return false; } The stuff inside the brackets (num == ""), is a logical test. If you're familiar with maths, it's essentially asking if what is inside the brackets makes mathematical sense. If not, it's asking if "" is the variable 'name'.

If this test is valid (ie. num is in fact equal to "" (Nothing)) then the second line is executed, forcing the function to stop and returning false.

If the statement does not make sense, then the second line is ignored.

Another example of a Select statement is 'if else'. Look at the statement at (2): ```JavaScript

else if (name.length <= 1) { // Checks if the name is longer that one character (2)
  return false;
}

```
```

This statement is exactly the same as an if statement, with the exception that it will only execute if the if statement in the line above fails the test.

Here is a screenshot example of a select statement in JavaScript.

```
if (counter >= pickaxeprice) { // If they have enough
    pickaxes = pickaxes + 1; // Give them a pickaxe
    pickaxedur = pickaxedur + pickaxedurnum;
    counter = counter - pickaxeprice; // Charge them for it
    pickaxeprice = Math.round((pickaxeprice * 1.5)*100)/100;// Increase and round
```

Part 3 - Explaining an Iteration Statement

A iteration statement is a statement that cycles through data or executes multiple times. The best example of a iteration statement is a for loop through an array.

```JavaScript

var names = ['Bob','Bill','John','James'];// defining a variable
for (var i = 0; i < names.length; i++) {// Creating to loop where i is any number //between 0 and the length of names
  alert('Hello ' + names[i]);// This will alert the user with every name
}

```

As you can see, the for loop cycles through items in the array 'names', and alerts for each name.

Here's an example of iteration in Python.

```
import datetime
days = ['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']
i = 0
for day in days:# iterate through days
    if datetime.datetime.today().weekday() == i:# if the day is today
        print "Today is " + day# print the day
    else:
        print "Today is not " + day# else, print that it's not the day
    i += 1
```

Part 4 - Influential Factors

When writing a program, it is important to select a language that suits your requirements.

Customer Requirements

An important factor in selecting the correct language is to see whether it is supported on a platform that the end user has convenient access to.

Organizational Policy

As a programmer, it is your job to make your code both functional, and readable. The organizational policy must be as simple as possible in order to make you code easy to access and expand upon.

Suitability in Terms of Available Features and Tools

Probably the most important factor in deciding a language is whether or not it contains the tools to do what you want quickly and efficiently. An example of this is popular integration with a particular task. PHP for example is widely supported and used in webservers, so it is a bit language to consider when writing web applications.

Experience / Availability of Trained Staff

Sometimes, it must be considered whether or not the staff currently employed are fluent in a particular language. Although it doesn't take long to learn a new language, small tasks present themselves where familiarizing yourself or teaching others a new programming language would take longer that writing the code in a previously known language.

Reliability

Reliability in terms of code is often referred to in terms of the compilation or interpretation of the code. For example, a low level language like C, would be reliable to execute extremely low level tasks, and the execute them efficiently, but due to this low level operation, it becomes inefficient to write C code that would be reliable on a high level. On the other hand, for high level tasks, languages such as Python are much less likely to make arithmetic mistakes. Although you lose the ability to manipulate data on a processor level, you gain the efficiency of having the interpreter do all of the data validation for you.

Development and Maintenance Costs

Some language frameworks require licensing costs. These costs are for proprietary libraries and such that are needed to run and develop software that incorporate software that has been predefined by the libraries. And example of this is Visual Studio, a software developed around a Windows proprietary graphical display framework.

Other maintenance costs include mainly servers and other hardware that are required to run the software, and the regular maintenance of those servers.

Expandability

Expandability of code is important in a corporate environment as your code may be expanded by anyone in the software department. The expandability of languages could include that libraries that it is compatible with. It could also include the readability of the code, as for someone to expand the code, they must first understand it. Most languages are expandable, but the way which you program may not be. It is sometimes of priority to write a program core, and a system for

implementing modules. This way you can distribute the completion or improvement of each individual module to a different person, and that person would only need to know the data that is inputted into the module, what the module is supposed to output, and the data types of that data.

Apendix

Here is an example of sequence in C#. (Just a side note, asking people to screenshot code is useless and awkward for everyone.)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            double num;
            if (textBox1.Text != "" && textBox2.Text != "" && double.TryParse(textBox2.Text, out num) && double.TryParse(textBox1.Text, out num))
            {
                long num1 = (long)Convert.ToDouble(textBox1.Text);
                long num2 = (long)Convert.ToDouble(textBox2.Text);
                long answer = num1 + num2;
                label1.Text = answer.ToString();
            }
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }
    }
}
```