## Scripting Language

**Object Oriented**

An object oriented programming is a program that is made up of classes and objects which involves defining objects and interacting with objects in order to enable a programmed action, for example, a game.

**Event Driven**

Event driven programming is a program that controls the flow of the program which is determined by events. These events reacts and responds to the user’s actions such as mouse clicks and button clicks.

**Procedural**

Procedural programming is a method of programming that makes use of procedures which is also known as subroutines. Procedural programming is a set of instructions which tells the computer what to do step by step.

## Classes and Objects

**Object**

Scripting language can create many different objects as they are object oriented. An object is a data which knows it properties and how it works. There are many different types of objects that are used in scripting language. The following below is an example of the String object:

<script>

hw=”Hello world”

document. write( hw.length )

</script>

**Properties**

Properties in OOP are a type of class member that can be used to set the property of an object or get the property of an object. For example, in OOP the class font could have the properties size and colour. The properties can be read using the getter method for the property and the property can be set using the setter method for the property.

**Methods**

Scripting language uses methods as it enables them to use common pieces of code which have already been used. Objects use methods in order to create the actions, which are then executed by the objects. The following below is an example of the String object:

<script>

hw=”Hello world”

document. write( hw.toUpperCase( ) )

</script>

## Javascript into HTML

Describe ways for placing JavaScript into HTML

**Internal**

JavaScript can be placed into the <body> and <head> sections of the HTML page. It is also important for the JavaScript code to be placed between <script> and </script> tags. The following below is an example of this:

<script>

document.getElementById("demo").innerHTML = "My First JavaScript";

</script>

**External**

JavaScript can be placed in external files and they have the file extension .js. External scripts are also very useful when you use the same code in different webpages. In order to use an external script you need to place the name of the script file in the source (src) attribute of a <script> tag as you can see from the example below.

<!DOCTYPE html>

<html>

<body>

<script src="myScript.js"></script>

</body>

</html>

## Hiding JavaScript from Old Browsers

JavaScript is hidden from old browsers because some old browsers do no support scripting language. JavaScript needs to be hidden from the old browsers in order to prevent confusion; this can be accomplished by placing the script within the HTML comments as you can see from the example below.

<script>

<!- -

Insert script here

//- -!>

</script>

## JavaScript Events

**Interface Actions**

Interface event actions are actions that are not a result of a user action but as a result of a user action. For example if the user clicks on a hyperlink on the page, this triggers the click action and the result of the click action could be to unload the current page and load a new page. The following below are some examples of interface event actions:

* Unload
* Resize
* Scroll
* Focus/blur

**Mouse Events**

The JavaScript mouse events runs whenever the user performs an action using the mouse. The following below are some common examples of JavaScript mouse events:

* onClick
* onMouseDown
* onMouseUp
* onMouseOver

**Form Events**

The JavaScript form events runs whenever the user performs an action inside a HTML form. The following below are some common examples of JavaScript form events:

* onSubmit
* onChange
* onFocus
* onBlur

**DOM**

DOM, which stands for Documented Object Model, is programming interface for HTML and XML documents. A DOM event is when an object is addressed.

**Microsoft Events**

Microsoft events have created many of their own events, which are only supported using the Microsoft Internet Explorer browser. The following below are some common examples of JavaScript Microsoft events:

* Right Click
* XML load
* MouseEnter
* MouseLeave

## JavaScript Security

When using JavaScript there are many security issues. For example, when the code is being implemented on the user’s computer it gives hackers the opportunity to gain access to the code. However, there are many different solutions to prevent JavaScript from being a security risk on websites. The following below are the things that JavaScript cannot do in order to keep JavaScript secure:

* JavaScript cannot read or write to clients (apart from cookies)
* JavaScript cannot run other programs.
* JavaScript are only able to interact with webpages from the same website.
* JavaScript cannot upload without permission which comes in form of an alert message
* JavaScript cannot read or view the browser history