

File Edit View History Bookmarks Tools Help

CIS120/17 Course Page

www.pgrocer.net/PGGCIS120.html

Programming: Logic, Design and Implementation
CIS120/17

About This Course:
Need help or support?
Read Me First
Introduction to CIS120/17
Syllabus
Withdrawal Policy as of Fall 2017

Site Resources:
Notes and Handouts
Examples
Programs
Presentations
SmartBoard Presentations
Audio lectures

Weekly Schedule:
Weekly schedule
Assignment summary

Links:
Links & Tutorials

The information at this site is in draft form and is the property of Priscilla Grocer. The information may not be duplicated or distributed without her permission. © Copyright 2019

Programming: Logic, Design and Implementation (CIS120/17)

Weekly Schedule

If when I return an assignment you do not get a grade it means that you have to go over the assignment and fix and resubmit. Remember, all assignments including inclass assignments must be done by everyone. There is no makeup on quizzes and tests (some tests are accepted late - check).

If you get an assignment back without a grade you need to fix it and resubmit to get a grade.

Schedule by week or unit	Information to cover
HELP schedule	<p>Please note that the first assignment is at the bottom of the list posted here. You should do the work in order! I may add to the weekly schedule during the course of the week, so please check back multiple times. Please keep copies of all work you submit until you receive your final grade at the end of the semester. Play computer means to figure out the results yourself by evaluating the data rather than running the query. Remember that I tape all my classes and capture notes on the Smartboard in the spring and fall. The recordings (audio) and the Smartboard are available for summer students to use under Audio and Smartboard.</p> <p>Joe Fletcher is my SI and he is available for help many hours during the week. Monday 2-7 in K104 Tuesday 12:30 - 4 in K104 Wednesday 12-3 in K130 and 3-6 in K104 Thursday 12:30 - 5 in K103 He can also do time on Friday morning if people let me know they would be able to use it. If you cannot find Joe, ask at the help desk in the middle of K building. They will know where he is</p>
Week #7 Week of March 2nd	<p>We will continue with JavaScript this week. We will work through the JavaScript we started last week. It is located at: JavaScript Examples Check back!</p>
Week #6 Week of February 24th	<p>I will not be on campus Thursday, February 27th. It is an online day. Joe will be available for help during class time. Check back! We will continue with the logic shown in last weeks writeup. We will go over the presentations on if statements and loops. The plan is to also look at JavaScript and we did. We started looking at the if statement. The JavaScript is under programs. If you have the book I suggest reading chapters #2 and #3. If you do not have the book continue with the articles I suggested in week 4 and throughout the course. Actually I think everyone should read the articles. Basic video Assignments: Flowchart and Pseudocode test This is a test. I will accept it late, but late points will be subtracted. As always it is due the Thursday of the week after it was assigned - so in this case March 5th. First Java Script If you are having any doubts or questions about JavaScript, I strongly urge you to come to class Thursday where you can get questions answered by Joe.</p>
	<p>We will continue with logic and concepts this week. Practice exercise: Practice exercise Practice continued Be sure to look at the solutions to the practice and I also recommend listening to the accompanying audio. Presentation on using pseudocode to play computer Separate speaker notes to accompany presentation on using pseudocode to play computer For help on if statements, look at this presentation: Presentation on logical if structures Separate speaker notes to accompany presentation on logical if structures</p>

10:56 AM
3/3/2020

The image shows a browser window displaying the source code of a JavaScript calculator. The code is as follows:

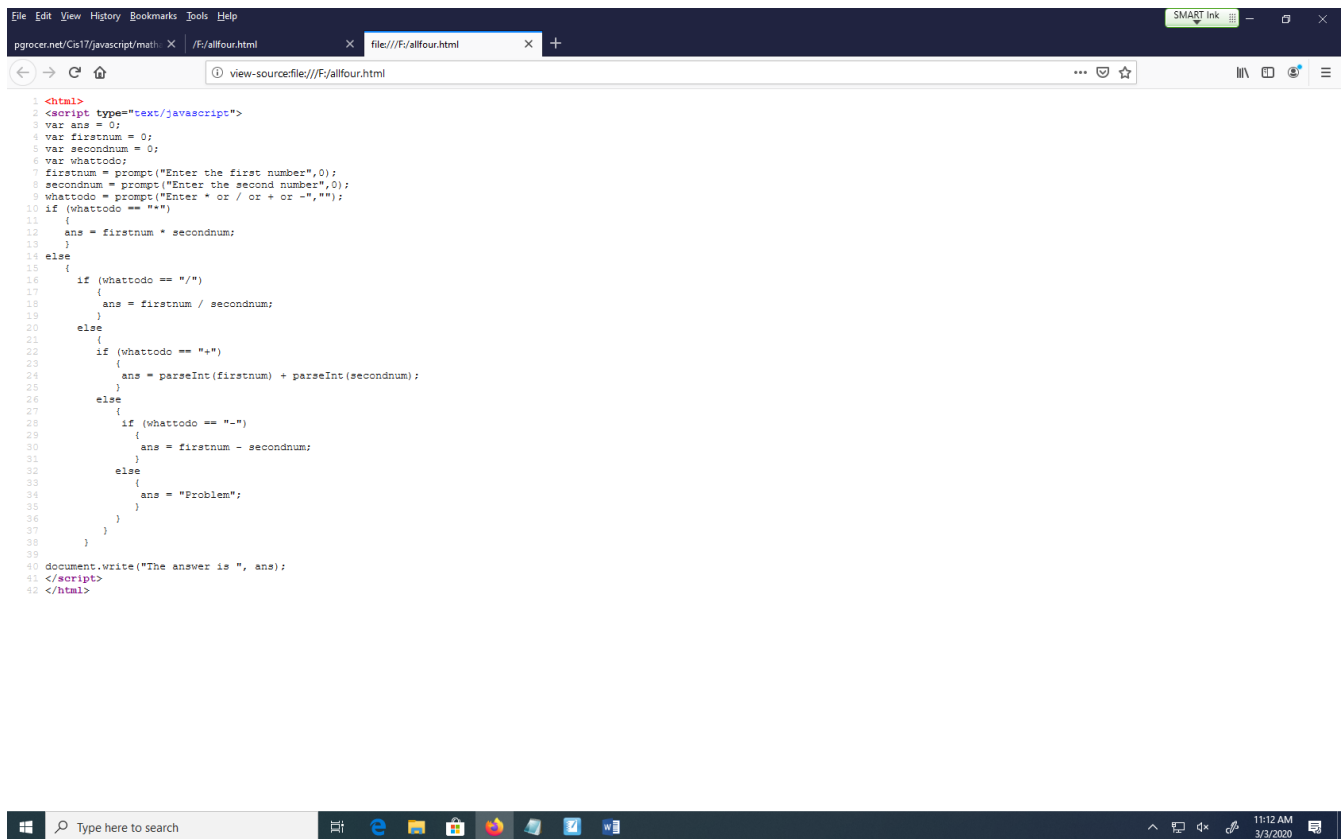
```
<html>
<script type="text/javascript">
var ans = 0;
var firstnum = 0;
var secondnum = 0;
var whattodo;
firstnum = prompt("Enter the first number",0);
secondnum = prompt("Enter the second number",0);
whattodo = prompt("Enter * or / or + or -","");
if (whattodo == "**")
{
ans = firstnum * secondnum;
}
else
{
ans = firstnum / secondnum;
}
document.write("The answer is ", ans);
</script>
</html>
```

Overlaid on this is a Notepad window containing the same code, but with several lines of code wrapped in curly braces and annotated with green handwritten marks:

```
var firstnum = 0;
var secondnum = 0;
var whattodo;
firstnum = prompt("Enter the first number",0);
secondnum = prompt("Enter the second number",0);
whattodo = prompt("Enter * or / or + or -","");
if (whattodo == "**")
{
ans = firstnum * secondnum;
}
else
{
if (whattodo == "/")
{
ans = firstnum / secondnum;
}
else
{
if (whattodo == "+")
{
ans = parseInt(firstnum) + parseInt(secondnum);
}
else
{
if (whattodo == "-")
{
ans = firstnum - secondnum;
}
else
{
ans = "Problem";
}
}
}
}
}
```

Handwritten annotations include:

- A large green 'F' on the left side of the Notepad window, spanning the entire conditional logic block.
- A green 'T' next to the opening curly brace of the main if statement.
- A green 'T' next to the opening curly brace of the division condition.
- A green 'T' next to the opening curly brace of the addition condition.
- A green 'F' next to the opening curly brace of the subtraction condition.
- A green 'F' next to the opening curly brace of the 'Problem' condition.
- Vertical green lines connecting the closing curly braces of the nested conditions back to the main if statement's closing brace.



```
1 <html>
2 <script type="text/javascript">
3 var ans = 0;
4 var firstnum = 0;
5 var secondnum = 0;
6 var whattodo;
7 firstnum = prompt("Enter the first number",0);
8 secondnum = prompt("Enter the second number",0);
9 whattodo = prompt("Enter * or / or + or -","");
10 if (whattodo == "**")
11 {
12     ans = firstnum * secondnum;
13 }
14 else
15 {
16     if (whattodo == "/")
17     {
18         ans = firstnum / secondnum;
19     }
20     else
21     {
22         if (whattodo == "+")
23         {
24             ans = parseInt(firstnum) + parseInt(secondnum);
25         }
26         else
27         {
28             if (whattodo == "-")
29             {
30                 ans = firstnum - secondnum;
31             }
32             else
33             {
34                 ans = "Problem";
35             }
36         }
37     }
38 }
39
40 document.write("The answer is ", ans);
41 </script>
42 </html>
```

```
1 <html>
2 <script type="text/javascript">
3 var myPoints;
4 var msg;
5 myPoints = prompt("Enter the number of points you have earned",0);
6 if (myPoints < 10)
7 {
8   msg = "Not enough points for a prize";
9 }
10 else
11 {
12   if (myPoints <= 50)
13   {
14     msg = "You can choose a prize from group B";
15   }
16   else
17   {
18     msg = "You can choose a prize from group A";
19   }
20 }
21 document.write(msg);
22 </script>
23 </html>
24
```

```
graph TD
  Start([START]) --> Declare[declare var]
  Declare --> Prompt[/prompt/]
  Prompt --> Cond1{myPoints < 10}
  Cond1 -- N --> Cond2{<= 50}
  Cond1 -- Y --> NoPrize[no prize]
  Cond2 -- N --> A[A]
  Cond2 -- Y --> B[B]
  A --> Merge(( ))
  B --> Merge
  NoPrize --> Merge
  Merge --> Write[/Write/]
  Write --> End([END])
```

The screenshot shows a web browser window with the source code of a JavaScript program. The code is as follows:

```
1 <html>
2 <head>
3 <title>JavaScript guess number game</title>
4 </head>
5 <body>
6 <script type="text/javascript">
7   var theName = window.prompt("Enter your name","");
8   var theState = window.prompt("Enter the state","");
9   if (theState == "MA" || theState == "RI")
10    {
11     document.write(theName + " you live in " + theState, "<br>");
12    }
13  else
14  {
15    document.write(theName + " you do not live in MA or RI <br>");
16  }
17 </script>
18 </body>
19 </html>
```

Handwritten annotations in blue and green ink are present over the code and to its right. The text "|| OR" is written in blue. Below it, "could have + " is written in green. To the right of the code, a logic diagram is drawn in blue ink. It consists of two diamond-shaped nodes labeled "MA" and "RI". From the "MA" diamond, a line labeled "N" goes to a box labeled "do not live in", and a line labeled "Y" goes to a box labeled "You live in". From the "RI" diamond, a line labeled "N" goes to the same "do not live in" box, and a line labeled "Y" goes to another box labeled "You live in". A green oval encircles the two "You live in" boxes, with the word "Same" written in green next to it. A green arrow points from the text "could have + " to the "MA" diamond.

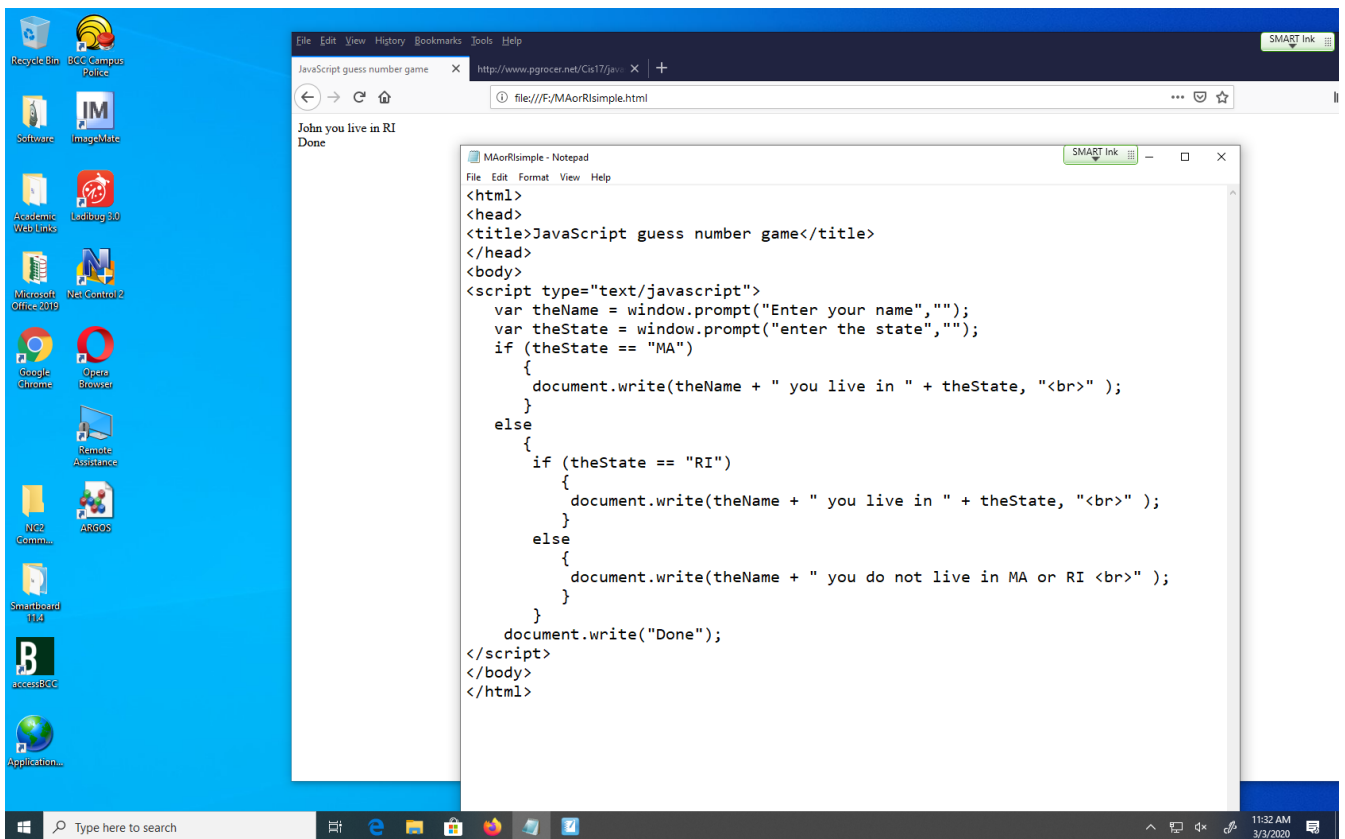
The browser's taskbar at the bottom shows the Windows logo, a search bar with the text "Type here to search", and several application icons including Edge, File Explorer, and Word. The system tray on the right shows the time as 11:23 AM on 3/3/2020.

The screenshot displays a SMART Notebook interface with two main components: a code editor on the left and a flowchart on the right.

Code Editor: The code is written in HTML and JavaScript, implementing a simple state-guessing game. It prompts the user for their name and state, then checks if the state is "MA" or "RI".

```
html>
  head>
    title>JavaScript guess number game</title>
  head>
  body>
    script type="text/javascript">
      var theName = window.prompt("Enter your name","");
      var theState = window.prompt("enter the state","");
      if (theState == "MA")
      {
        document.write(theName + " you live in " + theState, "<br>" );
      }
      else
      {
        if (theState == "RI")
        {
          document.write(theName + " you live in " + theState, "<br>" );
        }
        else
        {
          document.write(theName + " you do not live in MA or RI <br>" );
        }
      }
    script>
  body>
html>
```

Flowchart: A hand-drawn flowchart in blue ink illustrates the logic of the code. It starts with a decision diamond labeled "MA". If the answer is "N" (No), it leads to another decision diamond labeled "RI". If the answer to "RI" is "Y" (Yes), it leads to a box labeled "You live in". If the answer to "RI" is "N", it leads to a box labeled "do not live in". If the answer to "MA" is "Y", it also leads to a box labeled "You live in". A green oval encircles the two "You live in" boxes, with the word "Same" written next to it. Above the flowchart, the text "OR" and "could have + " is written in green.



JavaScript guess number game

Ann you meet the criteria

```
1 <html>
2 <head>
3 <title>JavaScript guess number game</title>
4 </head>
5 <body>
6 <script type="text/javascript">
7   var theName = window.prompt("Enter your name","");
8   var yrBirth = parseInt(window.prompt("enter the year of birth",0));
9   var marStat = window.prompt("enter your marital status","");
10  if (yrBirth > 1980 && marStat == "M")
11  {
12    document.write(theName + " you meet the criteria <br>");
13  }
14  else
15  {
16    document.write(theName + " you do not meet the criteria <br>");
17  }
18 </script>
19 </body>
20 </html>
```

Handwritten notes:

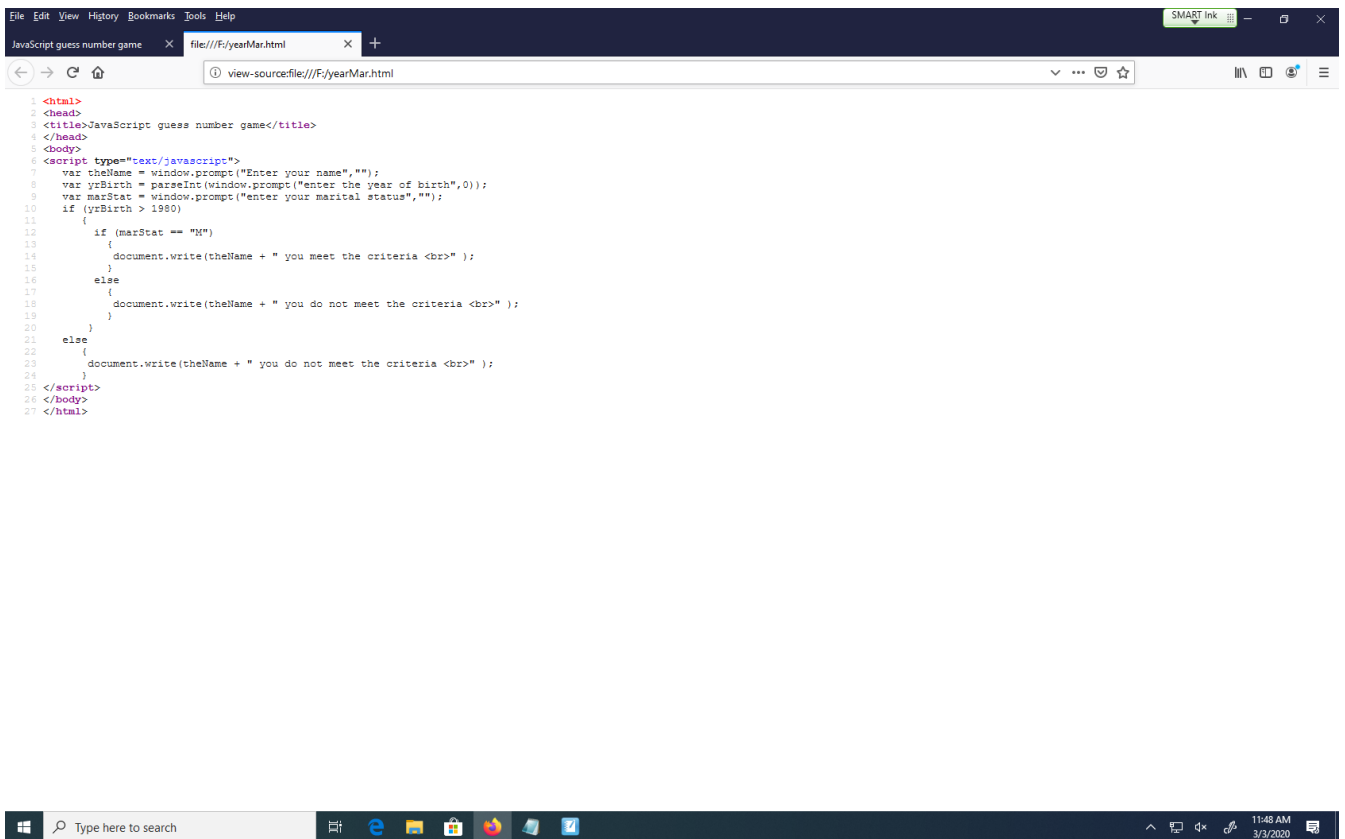
- AND**
- OR**
- YrB** Y F N N
- mar** Y N Y N
- meet / not meet
- MA** Y F N N
- RI** Y N Y N
- error because testing same field
- MAR** Meet
- not meet
- YrB > 1980** (Decision diamond)
- mar = M** (Decision diamond)
- not meet / Meets (Outcomes)
- same** (Note on flow)
- 1990 M** (Input example)
- 28 and** (Note)

The screenshot displays a SMART Notebook window with two main components:

- Code Editor (Left):** A Notepad window titled "yearMar - Notepad" containing the following JavaScript code:

```
<html>
<head>
<title>JavaScript guess number game</title>
</head>
<body>
<script type="text/javascript">
  var theName = window.prompt("Enter your name","");
  var yrBirth = parseInt(window.prompt("enter the year of birth",0));
  var marStat = window.prompt("enter your marital status","");
  if (yrBirth > 1980)
  {
    if (marStat == "M")
    {
      document.write(theName + " you meet the criteria <br>");
    }
    else
    {
      document.write(theName + " you do not meet the criteria <br>");
    }
  }
  else
  {
    document.write(theName + " you do not meet the criteria <br>");
  }
</script>
</body>
</html>
```

Handwritten annotations in green and red ink are present on the code, including a large 'T' and 'F' on the left side of the first conditional block, and a large 'F' on the left side of the second conditional block.
- Flowchart (Right):** A hand-drawn flowchart in red ink illustrating the logic of the code. It starts with a decision diamond "YrB > 1980". If "N" (No), it leads to a box labeled "not meet". If "Y" (Yes), it leads to another decision diamond "mar = M". From "mar = M", a "Y" leads to a box labeled "meets", and an "N" leads to a box labeled "not meet". A green oval encircles the "not meet" boxes from both the "YrB > 1980" and "mar = M" diamonds, with the word "same" written in green next to it. To the left of the flowchart, there are handwritten notes: "N", "N", and "not meet" with arrows pointing towards the flowchart.



File Edit View History Bookmarks Tools Help

Programs for CIS120/CIS17

www.pgrocer.net/Cis17/cis17programs.html

Programs for CIS120/CIS17 - Programming: Logic, Design and Implementation

Visual Basic (2010/2012/2015)	<p>firstSp12.zip - introduction to using VB2010/2012</p> <p>FirstMath.zip - calculations</p> <p>basicMath.zip - calculations</p> <p>loopswhile120.zip - introduction to loops</p> <p>workareas.zip - introduction to variables</p> <p>userInput.zip - introduction to user input</p> <p>deptArray.zip - introduction to arrays</p> <p>NextVB.zip - two programs in this zip</p> <p>VBFI1CIS120.zip - multiple programs</p> <p>projWriteText.zip - creating a file</p> <p>readProj.zip - reading a file</p>
JavaScript	<p>hello.html</p> <p>multiply.html</p> <p>multiplies.html</p> <p>addnum.html</p> <p>addnumcont.html</p> <p>mathans.html</p> <p>ifwithif.html</p> <p>MAorRI.html</p> <p>prob2.html</p> <p>whileloop.html</p> <p>doLoop.html</p> <p>ifwithloop.html</p> <p>finaltotals.html</p> <p>Loop comparison:</p> <p>Math facts while loop</p> <p>Math facts nested while loop</p> <p>Math facts do...while loop</p> <p>Math facts nested do...while loop</p> <p>Math facts for loop</p> <p>Math facts nested for loop</p> <p>Arrays:</p> <p>deptArray.html</p> <p>thedata.html</p> <p>thedata12.html</p> <p>thedata12another.html</p> <p>thedata11.html</p> <p>soupararray1.html Search using JavaScript</p> <p>Guess number, one guess</p> <p>Guess number, one game</p> <p>Guess number, multiple games</p>
	<p>helloWorld.html (not HTML5)</p> <p>helloWorld5.html</p> <p>multiply5.html</p> <p>addnum5.html</p>

Program/script language

- 1) Sequence
- 2) decisions if
- 3) ~~repetition~~ loop
Repetition

Type here to search

11:49 AM 3/3/2020

The screenshot shows a web browser window displaying the output of a JavaScript program. The output consists of four lines: "This is number 1", "This is number 2", "This is number 3", and "This is number 4", followed by "This is the end!!!".

Overlaid on the browser is a Notepad window containing the following JavaScript code:

```
<html>
<script type="text/javascript">
var data_input = 0;
data_input=prompt("How many times should I write the line?",0);
var ct=1;
while (ct <= data_input)
{
  document.write("This is number " + ct + "<br>");
  ct = ct + 1;
}
document.write("<br>" + "This is the end!!!")
</script>
</html>
```

Handwritten annotations in red ink include:

- A red "4" next to the prompt function.
- A red bracket on the left side of the while loop.
- The text "loop has a Control" written in red.
- A list of three steps in red: "1) set control", "2) check control", and "3) change control".
- Red mathematical expressions: $ct = 1$, $ct \leq \text{data_input}$, and $ct = ct + 1$.

Handwritten annotations in green ink include:

- The text "Control loop" written vertically.
- The text "ct total user input" written vertically.

The image shows a Windows desktop with two Notepad windows side-by-side. The left window contains JavaScript code for a while loop, and the right window contains code for a do-while loop. Handwritten red annotations and flowcharts are present on both windows.

Left Window (while loop):

```
<html>
<script type="text/javascript">
var data_input = 0;
data_input=prompt("How many times should I write the line?",0);
var ct=1;
while (ct <= data_input)
{
  document.write("This is number " + ct + "<br>");
  ct = ct + 1;
}
document.write("<br>" + "This is the end!!!")
</script>
</html>
```

Right Window (do-while loop):

```
<html>
<script type="text/javascript">
var data_input = 0;
data_input=prompt("How many times should I write the line?",0);
var ct=1;
do
{
  document.write("This is number " + ct + "<br>");
  ct = ct + 1;
}while (ct <= data_input)
document.write("<br>" + "This is the end!!!")
</script>
</html>
```

Handwritten Annotations:

- Left Window:** A red circle is drawn around the condition `ct <= data_input` in the while loop. A red arrow points from the text "No lines" to this circle. Below the code is a flowchart for a while loop: a diamond labeled "Check" has an "IN" arrow on the left and an "Out" arrow on the bottom. An arrow labeled "Y" goes from the top of the diamond to a rectangle labeled "Process", which then loops back to the top of the diamond. An arrow labeled "N" goes from the bottom of the diamond to "Out". Next to it is the text "may not do loop at all".
- Right Window:** A red circle is drawn around the condition `ct <= data_input` in the do-while loop. Below the code is a flowchart for a do-while loop: a rectangle labeled "Process" has an arrow going down to a diamond labeled "Check". An arrow labeled "Y" goes from the top of the diamond back to the top of the "Process" box. An arrow labeled "N" goes from the bottom of the diamond to "Out". To the right of this flowchart is the text "always does at least once".

The screenshot shows a web browser window with the URL `www.pnrocer.net/Cis17/javascript/withloop.html`. The browser displays the message "You can choose a prize from group A". Overlaid on the browser is a Notepad window containing the following JavaScript code:

```
<html>
<script type="text/javascript">
var inputPoints;
var totalPoints = 0;
var msg;
var ct = 1;
while (ct <= 5)
{
  alert("ct at beginning of loop " + ct);
  inputPoints = prompt("Enter the number of points you have earned",0);
  totalPoints = parseInt(totalPoints) + parseInt(inputPoints);
  alert("totalPoints after add " + totalPoints);
  ct = parseInt(ct) + 1;
}
if (totalPoints < 10)
{
  msg = "Not enough points for a prize";
}
else
{
  if (totalPoints <= 50)
  {
    msg = "You can choose a prize from group B";
  }
  else
  {
    msg = "You can choose a prize from group A";
  }
}
document.write(msg);
</script>
</html>
```

Handwritten in red on the right side of the Notepad window is the word "accumulate" and a diagram showing the calculation: $\text{total} + \text{input}$ over total , with a large red bracket on the right side of the equation.

The screenshot shows a web browser window displaying a tutorial for the JavaScript `String.toUpperCase()` method. The browser's address bar shows the URL `https://www.w3schools.com/jsref/jsref_touppercase.asp`. The page has a dark navigation bar with links for various programming topics like HTML, CSS, JavaScript, SQL, Python, PHP, Bootstrap, etc. A sidebar on the left lists JavaScript methods, with `toUpperCase()` highlighted. The main content area features a 'Get Started Today' button, a 'Previous' button, and a 'Next' button. The 'Example' section shows a code snippet:

```
var str = "Hello World!";  
var res = str.toUpperCase();
```

 Below this is a 'Try it Yourself' button. The 'Definition and Usage' section explains that the method converts a string to uppercase letters and includes a note that it does not change the original string. The 'Browser Support' section contains a table:

Method	Chrome	Edge	Firefox	Safari	Opera
<code>toUpperCase()</code>	Yes	Yes	Yes	Yes	Yes

On the right side of the page, there is a blue sidebar for an O'Reilly eBook promotion, titled 'O'Reilly Experimentation eBook', with a sub-header 'New O'Reilly feature experimentation eBook' and a call to action: 'Learn how to build and manage an experimentation platform. Get your free eBook today!'.