# ITSE 2302 – Program 02

All of the assignments in this course must be uploaded to Bb **and** *published to your student website*. Use the numbered list below **and** the Common Requirements as a confirmation checklist.

- Complete the steps described at the Website Setup & Publishing link in the "Assignments" section in Bb.
- Use the course material located at
  - [HTML Tutorial](#) , [CSS Tutorial](#) , [JavaScript Tutorial](#)
- Program 02 can be completed using content from these tutorial sections:
  - HTML HOME – DOM Node Lists
- Develop the page/site <u>for your user</u> – Communicate with your user.

**Requirements (remember to identify the requirements by #*number*):**

Note: the requirements below refer to the index.html within the program-## folder unless otherwise stated.

1. The Program 02 link on your top-level page should open an index.html page in the program02 folder. Include an HTML Canvas with 'strokeText' displaying a welcome message.
2. Include 2 textareas and a button that calls a function to concatenate the text in the textareas and displays the result in a paragraph.
3. Demonstrate the use of the following CSS Combinators: descendant, child, adjacent sibling, and general sibling.
4. Use the HTML Local Storage API to store the values in the textareas from Requirement #1. Include buttons and functionality to store, retrieve, and remove the localStorage objects.
5. Include a textbox and a textarea. Button – calls a function to determine if the string in the textbox is found in the contents of the textarea. Naturally, message to the user if the string was or was not found.
6. Include 3 textboxes. User enters a string in the first textbox and a number in each of the other textboxes. Include 3 buttons that call functions that perform slice, substring, and substr operations and display the results.

7. Include 3 textboxes. User enters a string in the first textbox. User enters string to be replaced in second textbox. User enters new string in third textbox. That is, find locations of string 1 in string 3 and replace those instances with string 2. Then, convert final string to uppercase. Button – calls a function to perform the replacement and displays the results in all uppercase. For example:
    a. String 1: basketball
    b. String 2: snow skiing
    c. String 3: My favorite sport is basketball!
    d. Replace basketball with snow skiing
    e. Result: MY FAVORITE SPORT IS SNOW SKIING!
8. Include a textbox. User enters a string of flower types separated by "&". Button – calls a function that converts the string into an array of flower types and displays the array contents.
9. Use the HTML sessionStorage to record and display the number of times an image of a panda bear is clicked.
10. Include 2 textboxes for user to enter 2 numbers. Advise the user to enter 0 in the second textbox so you can display the results of division by 0. Advise the user to enter a string to see the result. The first operation is to see the result of division by 0. The second operation is to see the result of using a text string in a mathematical operation. Button – call the function and display the result.
11. Demonstrate use of the Number(), parseInt(), and parseFloat() methods. Output the results in an HTML <table> element.
12. Button – calls a function that calculates and displays 3 random numbers:
    a. One between 1 and 7 (inclusive)
    b. One between 33 and 255 (inclusive)
    c. One between 572 and 723 (inclusive)
13. Button – calls a function that takes 2 numbers entered by the user. The first number is the base and the second is the exponent. Function calculates the exponentiation operation and displays the results.

14. Use Date methods to set and display dates 4 ways – 2 "long dates" and 2 "short dates". Note that when used this way, the date object is assigned to a variable containing that date value, not the current date object. Also note that the dates will, by default, display the same even though set differently.

15. Demonstrate use of the setDate() & getDate(), setMonth() & getMonth(), and setFullYear() & getFullYear() methods. Output the results in an HTML <table> element.

16. User enters 7 names of planets which are stored in an array named "planetsArray" and displayed back to the user. Then, join the array into a string and display the string.

17. Remove the last 2 items from the end of the planetsArray and then ask the user to add 2 new items.

18. Remove the first item from the beginning of the planetsArray and then ask the user to add 1 new item to the beginning.

19. Add 1 item to the planetsArray starting at the third position (i.e. [2]). Do not delete any items. After adding the item, sort the array (include the compare function even though not required here).

20. Take the first 4 items of the planetsArray to create a new array. Naturally, display both.

21. User enters 2 numbers. Compare those 2 numbers and report back to the user if the numbers are equal or, if not equal, which is greater.

22. User enters a number:
    a. Test to ensure the entry is numeric and message to the user to reenter if not.
    b. use logical operator to determine if the number is within the range 3-55 inclusive
    c. use logical operator to determine if the number is outside of the range 77-99 inclusive

23. User enters a type of metal. Use if-else-if statements to test if the type is one of 4 possibilities. Include a default condition.

24. User enters an actor's name. Use a switch statement to test if the name is one of 4 possibilities. Include a default condition.
25. User enters an animal. Use a switch statement with common code and fall-through to determine if the animal flies, runs, or crawls.
26. Using the planetsArray, demonstrate a:
    a. For/In loop
    b. While loop
    c. Do/While

---

TEST – TEST – TEST your **website** to ensure the requirements are met.

1. Use the list above **and** the Common Requirements as a confirmation checklist.
2. Not meeting all requirements = 0 points for the assignment.
3. Complete the appropriate Program ## confirmation in Bb by the due date.
4. Upload your .zip file to Bb under the appropriate Program ## link by the due date.
5. Ensure your assignment functions correctly on your student website and that all requirements are identified by "#XX".
6. Remember to study for the exam. Distributing learning over time is far superior than panic-study in close proximity to the exam date.