

Exam Skill Checklist and Proposal

Name:

Score:

10% - Skills checklist

Your exam will be marked according to the specific skills you show in your project. You will get credit for that skill if it works naturally and purposefully as part of solving your program's task.

Please check off each skill that you are going to use in your programming exam. This part of the worksheet is intended to be an inventory so that you can see everything that is expected from your project.

These questions are answerable using one class of concentrated effort. Please complete this worksheet and submit to your teacher. Your exam project is due June 17th.

The optional skills are used to replace other skills in the same category that you do not show in your exam project.

10% - HTML

- Contains one or more linked images
- Contains a button
- Contains a link
- Contains a header
- Contains a paragraph
- Contains a line break
- Contains strong text
- Contains a span
- Contains a list of items
- Contains a table

Option: Contains an HTML canvas

Option: Contains Scalable Vector Graphics

Option: Your suggestion:

5% - Style/CSS (choose 5)

- Sets the family of text
- Sets the color of text
- Sets the size of text
- Sets the weight of text
- Sets the style of text
- Sets the width of an element
- Sets the positioning (eg: absolute) of an element
- Positions an element using measurements

Option: Sets opacity

Option: Sets CSS animation

Option: Your suggestion:

5% - Input

- The user can make things happen by clicking a button
- The user can make things happen by clicking an image
- The user can enter information into an input line
- The user uses keyboard control
- The program reads the time

Option: The user uploads a file or a link

Option: Uses drop-down menus

Option: Your suggestion:

10% - Variables

- Uses numerical variables
- Initializes a numerical variable as **0**
- Uses string variables
- Initializes a string variable as ""
- Uses an array
- Initializes an array as []
- Uses an object
- Initializes an object as { }
- Uses a two dimensional array
- Uses a complex data structure (eg: an array of objects)

Option: Uses three dimensional arrays

Option: Uses pointers (an ugly hack, but it is kind of possible)

Option: Your suggestion:

15% - Processing

- Uses a singular **if** statement (each time)
- Uses an **if** → **else if** → **else** chain
- Uses a **for** loop (each time)
- Uses a nested loop
- Uses time functions
- Uses string addition (each time)
- Uses a **split** function
- Uses a **join** function
- Uses regular math functions (+-*/)
- Uses rounding functions (**round**, **floor**, **ceil**)
- Generates random numbers
- Uses additional math (powers, trig, etc.)

Option: Uses other string functions

Option: Uses regular expressions

Option: Uses other array or object functions

Option: Your suggestion:

5% - Output

- Outputs using innerHTML
- Adds/removes/changes an image
- Changes the position of an element
- Changes the style of an element
- Uses animation

Option: Your suggestion:

20% - Programming structure (each skill has double weight)

- Uses a function
- avoids of global variables whenever possible (ask for permission)
- Uses an object constructor with a property
- Uses an object constructor with a method
- Uses object constructors and object-oriented programming whenever possible
- Uses separate functions/methods for initializing data
- Uses separate functions/methods for reading input
- Uses separate functions/methods for processing
- Uses separate functions/methods for outputting information

Uses a function pipeline

Option: Uses setInterval/setTimeout

Option: Uses recursion

Option: Your suggestion:

20% - Programming style & communication

- Comment for the author of the program
- Comment for the date started and dates modified, including what was modified
- Comment for describing the whole program
- Comment for describing each initialization function or method **//INIT :**
- Comment for describing each input function or method **//INPUT :**
- Comment for describing each processing function or method **//PROCESS :**
- Comment for describing each output function or method **//OUTPUT :**
- Descriptive comments for each function or method
- Descriptive comments for each loop
- Descriptive comments for each if statement

- Uses descriptive, full language variable names
- Uses descriptive, full language function and method names
- Uses camelCase for variable and function names
- Uses UpperFirst for object constructor names

- Avoids lines longer than 100 characters
- Creates sub-functions/methods if a function/method is over 20 lines
- Accurately indents code

- Orders program by: main → init → input → process → output
- Orders object constructors by: init → input → process → output
- Separates HTML clearly into head (meta/script/style) and body

Program questions

1. What will your program do?
2. What will your user interface look like?
3. What variables and data structures will you need to store information?
4. What will you have to set up during initialization?
5. What will you use for input?
6. Please list the processing functions/methods that will be required:
7. What will you do for output?
8. How will you use an object constructor in your program?