CS712/CS812
Project Phase 3
Spring 2016
Due Sunday April 10

Add objects to your Tscript system by adding the following features:

1. (50 points) Object initialisers and property accessors. When implementing object initialisers, you do not need to implement the definition of get/set function properties, and make the internal [[Prototype]] property of the new object be null. The "base" of a property access must be an object, otherwise throw a type error message string (or report a type error if you are not supporting exceptions). You do not need to implement prototypal inheritance for this work item. You also do not need to implement property attributes. All properties should be data properties that are enumerable, writable and configurable.

2. (10 points) The new operator and prototypal inheritance. When creating a new object, take the prototype for the new object from the "prototype" property of the object produced by the expression of the new operator. If the "prototype" property of the object does not exist or is not an object, then make the prototype of the new object be null. Do not report an error if the expression for the new operator is not a function object. (In fact, because some students may not have functions implemented, I will only test this level of the assignment with non-function values for the expression of the new operator.)

3. (10 points) The global environment and the global object. Support the three value properties of the global object: NaN, Infinity and undefined. (Because we are not implementing property attributes, it is okay for the properties of the global object to be configurable, enumerable and writable.) Be sure to change the behavior for assigning to undefined identifiers to cause properties to be added to the global object.

4. (10 points) Built-in functions. This will require that you implement function call, if you have not already done so. Support a non-standard function property of the global object called "readln" which reads the next line from stdin and returns it (after discarding the newline on the end) as a string. At EOF, the function should return null. Also support these two standard function properties of the global object: isNaN and isFinite. Note that these functions take an argument, so your function call mechanism needs to support at least one argument.

5. (10 points) The **this** keyword. Add a non-standard function property of the global object called "testThis" that will expect an object to be bound to "this", will add a property to it called "xyz" containing the value 42, and will return "this". This function object should have a "prototype" property referencing an object that contains a function property called "printXYZ" that will display the contents of the "xyz" property of the object referenced by "this". I will test calling the "testThis" function as both a constructor and as a function.

6. (10 points) Add support for objects to the abstract operation ToPrimitive. This involves implementing the "internal" method DefaultValue. For this level, you will need to have function expressions implemented. And you should include support for references to "this" in the body of functions.

You should implement items in the above list in the order that they are listed. Items earlier in the list might be used to test items later in the list.

Remember that you need to be sure that your compiler supports the first four work items from Phase 1 (logical not operator, the equals operator, the less-than operator, the greater-than operator, the Boolean type, the Null

type and the Undefined type) and the first four work items from Phase 2 (the block statement, the empty statement, the if statement and the while statement).

To get full credit, your code must be adequately documented and structured. If I can't easily read and understand your code, you may lose points.

You must give me back the system in the same form that I gave it to you. I must be able to install it and run it in the exact same way as when it was delivered to you. I would also like you to keep the source code organized in the same directory hierarchy. If you fail to do this, a significant deduction will be made to your grade.

Note that there is a script, *distribute.sh*, in *tscript/bin*, which will generate a tar file for you. This tar file is what I require you to submit to me for grading.

To turn in this assignment, type:
~cs712/bin/submit phase3 tscript.tar

Submissions can be checked by typing:
~cs712/bin/scheck phase3

The assignment is due on Sunday April 10. Submissions between 8am April 11 and 8am April 12 will have a late penalty of 10 points. Submissions between 8am April 12 and 8am April 13 will have a late penalty of 30 points. **No program may be turned in after 8am on Wednesday April 13.**

Remember: you are expected to do your own work on this assignment.

Last modified on March 11, 2016.

Comments and questions should be directed to pjh@cs.unh.edu