**WAP – Assignment 7**

Contents

[**Task 1** 2](#_Toc39000164)

[**c function env** 2](file:///D:\2020\MIU\CS472-WAP\Assignments\w2d1\w2d1_Tamir_Batbayar.docx#_Toc39000165)

[**Global Env** 2](file:///D:\2020\MIU\CS472-WAP\Assignments\w2d1\w2d1_Tamir_Batbayar.docx#_Toc39000166)

[**Task 2** 3](#_Toc39000167)

[**Task 3** 3](#_Toc39000168)

[**Task 4** 4](#_Toc39000169)

[**myFunction env** 4](file:///D:\2020\MIU\CS472-WAP\Assignments\w2d1\w2d1_Tamir_Batbayar.docx#_Toc39000170)

[**Global Env** 4](file:///D:\2020\MIU\CS472-WAP\Assignments\w2d1\w2d1_Tamir_Batbayar.docx#_Toc39000171)

[**Task 5** 5](#_Toc39000172)

[**a function env** 5](file:///D:\2020\MIU\CS472-WAP\Assignments\w2d1\w2d1_Tamir_Batbayar.docx#_Toc39000173)

[**Global Env** 5](file:///D:\2020\MIU\CS472-WAP\Assignments\w2d1\w2d1_Tamir_Batbayar.docx#_Toc39000174)

**Tamir Batbayar**

**611144**

# **Task 1**

##### **//Result**

**undefined** //inside c function: x = undefined

**8**  //inside c function: a=8

**8**  //inside f function: b=8

**9** //inside c function: b=9

**10** //global env: b=10

**1** //global env: x=1

#### **f function env**

**Phase 1: Hoisting**

var x = undefined;

**Phase 2: Execution**

b = a; //b = 8

document.write(b); //3: **8**

b = c;

x = 5;

### **c function env**

**Phase 1: Hoisting**

var f = undefined;

var x = undefined;

**Phase 2: Execution**

document.write(x); //1: **undefined**

document.write(a); //2: **8**

f = function(a, b, c){…}

f(a, b, c); //f(8,9,10)

document.write(b); //4: **9**

x = 10;

## **Global Env**

**Phase 1: Hoisting**

var x = undefined;

var a = undefined;

var b = undefined;

var c = undefined;

**Phase 2: Execution**

x = 1;

a = 5;

b = 10;

c = function(a, b, c){…}

c(8, 9, 10);

document.write(b); //5: **10**

document.write(x); //6: **1**

# **Task 2**

Global scope is a declaration in outside of function scope. It can be accessible from all inner scopes /function scopes/ .

Local scope is a function scope which is only can use in function block and inner function scopes.

# **Task 3**

a) **No**

b) **Yes**

c) **No**

d) **Yes**

e) **Yes**

**Explanation:**

Inner functions access their outer declared variable expect don’t have the same named declared variable. JS engine run first hoist to find all global declaration and function’s name without expression. Then second execution, it’s run the expression row by row. Inner function scope it’s run as same 2 phases.

# **Task 4**

#### **//Result**

**81**

**25**

### **myFunction env**

**Phase 1: Hoisting**

//no hoisting

**Phase 2: Execution**

return x \* x;

## **Global Env**

**Phase 1: Hoisting**

var x = undefined;

myFunction = function(){…};

**Phase 2: Execution**

x = 9;

document.write(myFunction()); //81

x = 5; //x’s value is changed to 5

document.write(myFunction()); //25

# **Task 5**

#### **//Result**

**10**

### **bar function env**

**Phase 1: Hoisting**

var foo = undefined;

**Phase 2: Execution**

//below condition is true, //because undefined is falsey

if(!foo){

foo = 10;

}

alert (foo); //10

## **Global Env**

**Phase 1: Hoisting**

var foo = undefined;

bar = function(){…};

**Phase 2: Execution**

foo = 1;

bar();