ASSIGNMENT7

1.

- **Undefined** : because variable x hoisted to the top the function c but wouldn't be assigned until the last line of the function.
- **8:** a is local variable passed as a parameter to function c, and it Is assigned 8 when passed to the function.
- **8 :** b passed to function c will be passed to function f therefore the variable b will be assigned to the function scope of f
- **9:** variable b here belongs to function c, and value of b which is 9 is passed as a parameter to the function c.
- **10:** variable b here belongs to the global scope therefore b=10
- **1 :** variable x here belongs to the global scope therefore x=1;

2.

- **The Global scope is the scope outside of all function definitions within the JavaScript file**
- **Local scope is the scope for definitions of variables within a function or a block**

3.

A. **No they don't have access to variables B and c :** because the global scope don't have access to the local lexical environment
B. **Yeah they do:** because statements in the local lexical environment has access to the global environment outside of it
C. **No they don't.** because the outside function scope don't have access to the local lexical environment

D. **Yeah they do:** because statements in the local lexical environment has access to the environment outside of it

E. **Yeah they do:** because statements in the local lexical environment has access to the environment outside of it

4.

- **81 :** var X's firs declaration is 9 so when myFunction is called at first it squares 9
  25: Var X is re assigned to 5 so when we call myFuntion for the second time it squares 5.

5.

- **alert(foo) will print 10 since foo is hoisted and assigned as an undefined variable in the local scope of bar. The program will get into the if(!foo) function because foo being hoisted and having the value of undefined implies a falsy value , the negation of foo will give us true thus resulting the if condition being executed and hence 10 being assigned to foo.**