ASSIGNMENT-1

1.What will be the output of this code?

Input: console.log(x);

var x=5;

Code:

```
console.log(x);
var x=5;
```

Output: undefined

Explanation: The first console.log(x) outputs 'undefined' because var x is hoisted and declared, but not yet initialized.

After var x=5; then assigns the value 5 to x.

2.What will be the output of this code?

Input: console.log(a);

var a;

Code:

```
console.log(a);
var a;
```

Output: undefined

Explanation: The first console.log(a) outputs 'undefined' because var a is hoisted and declared, but not yet initialized.

3.What will be the output of this code?

Input: console.log(b);

b=10;

var b;

Code:

```
console.log(b);
b=10;
var b;
```

Output: undefined

Explanation: The first console.log(b) outputs 'undefined' since var b is hoisted but not yet initialized.

After b = 10 assigns the value 10 to b. The var b; redeclaration does nothing since b is already declared.0-6

4. What will happen here?

Input: console.log(c);

Code:

```
console.log(c);
```

Output: undefined

Explanation: The Code console.log(c); will result in a RefernceError as c is not defined. This error occurs because c has not been declared in the scope before being logged. =]

5. What will be the output of this code?

Input: console.log(e);

var e=10;

console.log(e);

e=20;

console.log(e);

Code:

```
console.log(e);
var e=10;
console.log(e);
e=20;
console.log(e);
```

Output:

undefined

10

20

Explanation: The first console.log(e); throws a ReferenceError and we get ouput 'undefined', because in js variable declarations are hoisted to the top of their scope, but they are not initialized until the point of assignment.

After var e=10; executes, e is now been initialized to the value 10, therefore second console.log(e); prints output 10.

After e=20; changes the value of e to 20, the third console.log(e); prints the output 20.

6. What will be the output of this code?

Input: console.log(f);

var f=100;

var f;

console.log(f);

Code:

```
console.log(f);
var f=100;
var f;
console.log(f);
```

Output:

Undefined

100

Explanation: The first console.log(f) outputs ‘undefined’. This is because var declarations are hoisted to the top and initialized to‘undefined’.

After var f=100; f is set to 100. The second var f; does nothing as f is already declared.

The Second console.log(f) outputs 100.

7.What will be the output of this Code?

Input: console.log(g);

var g=g+1;

console.log(g);

Code:

```
console.log(g);
var g=g+1;
console.log(g);
```

Output:

undefined

NaN

Explanation: The first console.log(g) outputs ‘undefined’. This is because var g is hoisted but not yet initialized.

Then var g=g+1; tries to add 1 to the g, but since g is undefined, the result is NaN(Not-a-Number).

The second console.log(g) outputs NaN because that’s the value of g after the operation.

8.What will be the output of this code?

Input: var h;

console.log(h);

h=50;

console.log(h);

Code:

```
var h;
console.log(h);
h=50;
console.log(h);
```

Output:

undefined

50

Explanation: The first console.log(h) outputs ‘undefined’ because h is declared but not yet initialized.

After h=50, the second console.log(h) outputs 50 because h is now set to 50.

9.What will be the output of this code?

Input: console.log(i);

i=10;

var i=5;

console.log(i);

Code:

```
console.log(i);
i=10;
var i=5;
console.log(i);
```

Output:

undefined

5

Explanation: The first console.log(i) outputs ‘undefined’ because the variable hoisted to the top and not yet initialized.

After var i=5, i is set to value 5

The seconde console.log(i) outputs 5, because now i is set to 5.