Task

1)What will be the output?

```
Let x=5;
Let y=x;
x=10;
console.log(x);
console.log(y);
```

Output;10,5

Explanation: First x starts at 5. Then y gets the value of x which is also 5. When x changes to 10, y still holds the original value of 5.

2)What will be the output?

```
Let obj 1= {name: “Alice”};
Let obj 2=obj 1;
Obj 1.name=” Bob”;
console.log (Obj 1.name);
console.log (Obj 2.name);
```

Output; Bob, Bob

Explanation: obj1 starts with the name Alice. When obj2 is set to obj1, they reference the same object. Changing obj1.name to Bob also updates obj2.name to Bob, so both print Bob.

3) what will be the output?

```
Let a=” hello”
Let b=47;
Let c=true;
Let d= {key: “value”};
Let e=null;
Let f=undefined;
console.log (type of a);
console.log (type of b);
console.log (type of c);
console.log (type of d);
console.log (type of e);
console.log (type of f);
```

output: string
number
Boolean
object
object
undefined

Explanation: The variable a is a string, so its type is string. The variable b is a number, so its type is number. The variable c is a Boolean, so its type is Boolean. The variable d is an object, so its type is object. Note that null is also considered an object in JavaScript. The variable e is null, which is also classified as object, and f is undefined, so its type is undefined.

Task

4) what will be the output ?

Let numbers= [10,20,30,40,50];
console.log (numbers [2]);
console.log (numbers [0]);
console.log (numbers [numbers. length-1]);

output: 30,10,50

Explanation: numbers [2] retrieves the third element, which is 30. numbers [0] retrieves the first element, which is 10. numbers [numbers. Length - 1] accesses the last element, which is 50.

5)Let fruits= [“apple”,” banana”,” mango”];
Fruits [2] =” orange”;
Console.log(fruits);

Output: apple, banana, orange

Explanation: The original array fruits contain apple, banana, and mango. The value at index 2 (which is
mango is replaced with orange. When printed, the array now shows apple, banana, and orange.

6) Let matrix= [
[1,2,3],
[4,5,6],
[7,8,9]
];
Console.log (matrix [1][2]);
Console.log (matrix [2][0]);

Output: 6,7

Explanation: matrix [1][2] accesses the element in the second row and third column, which is 6. Matrix [2][0] accesses the element in the third row and first column, which is 7.

7)let person= {
Name:” john”,
Age:25,
City: “New York”
};
Console.log(person.name);
Console.log (person. age);

Output: john,25

Explanation: Name accesses the value of the name property, which is john. Person. Age accesses the value of the age property, which is 25.

8)let car= {
make:” Toyota”,
model: “corolla”,
year: 2021
};
Console.log(car[“make”]);

Task

Console.log (car[“model”]);

Output: Toyota, corolla

Explanation: car["make"] retrieves the value of the make property, which is "Toyota. “Car["model"] retrieves the value of the model property, which is corolla.

9)let book= {
title:” The Great Gatsby”,
author: “F. Scott Fitzgerald”
};
Book. Author=” Anonymous”;
Console.log (book. Author);

Output: F. Scott Fitzgerald

Explanation: The line Book. Author = Anonymous; doesn't affect book because JavaScript is case sensitive, and Book is not defined. When console.log (book. Author) is called, it still shows the original
author, F. Scott Fitzgerald.

10)let student= {
name:” Alice”,
grade: “A”
};
student. age=20;
Console.log (student);

Output: name: 'Alice', grade: 'A', age: 20

Explanation: The object student initially has properties name and grade. A new property age is added to the student object with a value of 20. When console.log(student) is called, it shows all three properties: name, grade, and the newly added age.