1. Below is a JavaScript program that creates bank coount objects and stored in the array ***bankAccounts.*** Trace the ouput without using Visual Studio.

```
let bankAccounts = [];
// Create bank account objects using object literals with the bank's
//  name as account holder
let account1 = {
  accountNumber: '123456789',
  bank: 'Bank of America',
  balance: 5000,
  type: 'Savings'
};
let account2 = {
  accountNumber: '987654321',
  bank: 'Chase Bank',
  balance: 10000,
  type: 'Checking'
};

let account3 = {
  accountNumber: '456789123',
  bank: 'DBS',
  balance: 2500,
  type: 'Checkings'
};

let account4 = {
  accountNumber: '223489123',
  bank: 'OCBC',
  balance: 4500,
  type: 'Savings'
};
// Push the bank account objects into the bankAccounts array
bankAccounts.push(account1, account2, account3,account4);
//Part (a)
console.log(bankAccounts[1].type);
console.log(bankAccounts[0].balance + bankAccounts[2].balance);

//Part (b)
console.log("Balance of Savings Accounts:");
 for (let i = 0; i < bankAccounts.length; i++) {
  let account = bankAccounts[i];
  if (account.type == 'Savings') {
    console.log(account.balance);
  }
}
```

a) What is the output Part (a) of the program segment?

b) What is the output for Part (b) of the program segment?

c) Below is a segment of codes added to the program. What is the output?

```
  function getTotalCheckingsBalance(accounts) {
    let totalBalance = 0;
    for (let i = 0; i < accounts.length; i++) {
      if (accounts[i].type == 'Checkings') {
        totalBalance += accounts[i].balance;
      }
    }
    return totalBalance;
  }
  //Part (c)
  // Calculate and output the total balance of 'Checking' accounts
  let totalCheckingsBalance = getTotalCheckingsBalance(bankAccounts);
  console.log("Total Balance of Checking Accounts: " + totalCheckingsBalance);
  // end of program
```

Output :

d) Below is a segment of codes added to the program. What is the output?

```
let search ="OCBC";
function displayBankdetails(accounts) {
    let totalBalance = 0;
    let notFound = true
    for (let i = 0; i < accounts.length; i++) {
      if (accounts[i].bank == 'OCBC') {
        return accounts[i]      // returns an object
      }
    }
  }

  let account = displayBankdetails(bankAccounts);
   console.log(search + " is a " + account.type + " account, balance is $" +
 account.balance);
// end of program
```

Output :

You may open up your Visual Studio Code (VSC) and work from there for the remaining questions.

2. Use the codes segment in Exercise 1 which you did in Topic 8 lesson. You may change the values of the length and width.

```
let r1 = {length:1, width: 5};
let r2 = {length:2, width: 5};
let r3 = {length:3, width: 5};
let r4 = {length:4, width: 5};
let rectArray = [r1, r2, r3, r4];
```

a) Write a function getBiggestlength() that ;

- accepts one parameter ie array ***rectArray,***
- return the largest length for output in the **main program**

b) Write a function getBiggest() that ;

- accepts one parameter ie array ***rectArray,***
- return the largest length and the object for output in the **main program**

3. In the main program, declare an array named ***sqArray*** to store square objects. Use the random function to generate 10 random integer values (10 - 20) as their length and store them in ***sqArray***. Print out the length and area of all the elements in the array.

Hint : Use this to generate the random integer values

```
Math.floor(Math.random() * 11) + 10
```

Your main program should produce the following output:

```
Length       Area
15           225.0
11           121.0
19           361.0
19           361.0
12           144.0
11           121.0
19           361.0
17           289.0
13           169.0
10           100.0
```

Sample code template to help you get started:

```
// Function to generate random length for square


// Function to calculate area of a square


// Blank array to store squares


// Create squares with random lengths


// Calculate area for each square and display output
```

4. A JavaScript program is required to calculate the total material cost of 4 objects.

a) Create 4 shape objects : name them : circle, square , rectangle and triangle. Each object should have the following properties. Your may set the values to your preference e.g radius is 5.

Object name	**Properties**	**Formular to calculateArea**
circle	radius materialCostPerUnitArea	3.142 * radius * radius
square	length materialCostPerUnitArea	length * length
rectangle	length width materialCostPerUnitArea	length * width
triangle	Base, height materialCostPerUnitArea	0.5 * base * height

b) Each object should have a method called ***calculateArea*** that calculates the area of the object.

c) Write a separate function called ***calculateMaterialCost*** that takes :

- two parameters: area and materialCostPerUnitArea,
- return the total material cost by multiplying the area by the materialCostPerUnitArea.

d) Create an array called ***shapes*** and store the four shape objects in it.

e) In the main program:

- Use a for loop to iterate through the shapes array.
- For each shape, calculate the area and material cost using the methods and function defined.
- Use console.log to display the name of the shape, the calculated area, and the material cost.

Sample Output :

```
Circle
Area: 3.14
Material Cost: 31.42

Square
Area: 4.00
Material Cost: 60.00

Rectangle
Area: 10.00
Material Cost: 120.00

Triangle
Area: 8.00
Material Cost: 80.00
```

Sample code template to help you get started:

```
// Create the circle object

// Create the square object


// Create the rectangle object


// Create the triangle object


// Function to calculate the material cost


// Store object into an array of shapes

// main program that loops through the array of shapes using for loop and
// output the area and material cost
```

- END -