

NAME: DISHA SHAH

Question 1:

```
var employees = [  
  { id: 10, name: 'John', experience: 5},  
  { id: 15, name: 'Mathew', experience: 7},  
  { id: 20, name: 'keith', experience: 12},  
  { id: 25, name: 'louis', experience: 5},  
  { id: 30, name: 'mark', experience: 4},  
  { id: 35, name: 'Ash', experience: 3},  
  { id: 40, name: 'robert', experience: 6},  
]
```

Print: All ID's in an array example: [10, 15, 20, 25, 30, 35, 40]

```
console.log([employees[0].id,employees[1].id,employees[2].id,employees[3].id,employees[4].id,employees[5].id,employees[6].id]);
```

//or

```
const id=employees.map(employee=>employee.id);  
console.log(id);
```

Print: TOTAL no of years' experience they all have

```
var exp=employees.reduce((acc,current)=>{  
  return acc + current.experience;  
},0);  
console.log(exp);
```

Print: All employees whose experience is greater than 5 years

```
var y = employees.filter(employee=>employee.experience>5);  
console.log(y);
```

Question 2:

```
let cat = {  
  catName: 'Kitty',  
  sayName () {  
    console.log(this.catName);  
  }  
}
```

```
let sayName = cat.sayName;
```

NAME: DISHA SHAH

Fix to Print out 'Kitty'

```
let cat = {  
  catName: 'Kitty',  
  sayName () {  
    console.log(this.catName);  
  }  
}  
  
let sayName = cat.sayName();
```

Question 3:

```
class user {  
  displayName() {  
    return this.name + ' is in ' + this.location;  
  }  
}  
  
var user1 = new user('Marlabs', 'New Jersey');  
  
console.log(user1.displayName());
```

Print: 'Marlabs is in New Jersey' (DO NOT create variables outside the User class)

```
class user {  
  constructor(name, location){  
    this.name = name;  
    this.location = location;  
  }  
  displayName() {  
    return this.name + " is in " + this.location;  
  }  
}  
  
var user1 = new user("Marlabs", "New Jersey");  
  
console.log(user1.displayName());
```

Question 4:

```
function myName(fName, lName) {  
  console.log(`my name is ${fName} ${lName}`);  
  
  function fullDetails(companyName) {  
    const value = `my name is ${fName} ${lName} and I work for ${companyName}`
```

NAME: DISHA SHAH

```
    return value;
  }

  return fullDetails;
}
```

```
Print: 'My name is Althaf Pattan and I work for Marlabs'
var y = myName("Althaf", "Pattan").call(this, "Marlabs");
console.log(y);
```

Question 5:

```
var pokemon = {
  firstname: 'Pika',
  lastname: 'Chu ',
  getPokeName: function() {
    var fullname = this.firstname + ' ' + this.lastname;
    return fullname;
  }
};
```

```
var pokemonName = function(snack, hobby) {
  console.log(this.getPokeName() + 'I choose you!');
  console.log(this.getPokeName() + ' loves ' + snack + ' and ' + hobby);
};
```

Print: 'Pika Chu loves sushi and algorithms'

```
var pokemon = {
  firstname: 'Pika',
  lastname: 'Chu ',
  getPokeName: function() {
    var fullname = this.firstname + ' ' + this.lastname;
    return fullname;
  }
};
```

```
var pokemonName = function(snack, hobby) {
  console.log(this.getPokeName() + 'I choose you!');
  console.log(this.getPokeName() + ' loves ' + snack + ' and ' + hobby);
}.bind(pokemon);
```

```
pokemonName("sushi", "algorithm");
```

NAME: DISHA SHAH

Question 6:

```
var x = '5';
var promise1 = new Promise(function(resolve, reject) {
  if (x = 5) {
    let data = {
      name: 'Marlabs',
      address: {
        state: 'NJ',
        zip: 084444
      }
    }
    console.log(data);
    resolve();
  } else {
    var res = new Error('failed.....');
    reject(res);
  }
});
```

```
var promise2 = function(address) {
  var msg = address.state + ' ' + address.zip;
  return Promise.resolve(msg);
}
```

```
var myFunc = function() {
  promise1
  .then(promise2)
  .then((data) => {
    console.log(data);
  })
  .catch(err => {
    console.log(err.message);
  });
};
```

Print: 'Marlabs address: New jersey, 08444'

```
var x = 5;
var promise1 = new Promise(function(resolve, reject) {
  if (x == 5) {
    let data = {
      name: 'Marlabs',
      address: {
        state: 'NJ',
```

NAME: DISHA SHAH

```
        zip: 08444
      }
    }
    console.log(data.name);
    resolve(data.address);
  } else {
    var res = new Error('failed.....');
    reject(res);
  }
});

var promise2 = function(address) {
  var msg = address.state + ' ' + address.zip;
  return Promise.resolve(msg);
}

var myFunc = function() {
  promise1
  .then(promise2)
  .then((data) => {
    console.log(data);
  })
  .catch(err => {
    console.log(err.message);
  });
};

myFunc();
```