Assignment – 12
ES6, Node.js

1. **Define an arrow function to find the sum of digits of a given number.**

   Code:-

```
//Sunil Kumar Sahoo, 23MMCI38, 16
let add = (num)=>{
   let rem, sum = 0;
   while(num > 0){
      rem = num % 10;
      sum = sum + rem;
      num = parseInt( num / 10);
   }
   return sum;
}
let sum = add(2302);
console.log("Sum is", sum);
```

   Ouput:-

```
Sum is 7
```

2. **Define a JS lambda expression to perform multiplication of two numbers.**

   Code:-

```
//Sunil Kumar Sahoo, 23MMCI38, 16
let mul = (a, b) => a*b;
console.log(mul(5, 9));
```

   Output:-

```
45
```

3. **Define a function double which takes a number and a callback arrow function. The function doubles the number and the callback function displays the result.**

   Code:-

```
//Sunil Kumar Sahoo, 23MMCI38, 16
function double(num,callback){
   n=num*2;
   callback(n);
}
double(8,(data)=>{
console.log(data);
})
```

   Output:-

```
16
```

4. **Create an Employee class with the properties name, position, salary and a method getDetails() which will return a string containing all the informations. Extend the Employee class into a Manager class that includes: department and bonusPercentage as properties and methods like – calculateBonus(), calculate and returns the bonus amount and getManagerDetails(), returns a string containing the manager's details, including the department and bonus. Use constructor to initialize the data. Create object of both employee and Manager Class and display their details.**

Code:-

```
//Sunil Kumar Sahoo, 23MMCI38, 16
class Employee{
   constructor(name, position, salary){
      this.name = name;
      this.position = position;
      this.salary = salary;
   }
   getDetails=()=>{
      console.log("Name: "+this.name+"\nPosition: "+this.position+"\nSalary: "+this.salary);
   }
}
class Manager extends Employee{
   constructor(name, position, salary, department, bonus){
      super(name, position, salary);
      this.department = department;
      this.bonusPercentage = bonus;
   }
   calculateBonus=()=>{
      return this.salary + (this.bonusPercentage/100.0)*this.salary;
   }
   getManagerDetails=()=>{
      this.getDetails(this.name, this.position, this.salary);
      console.log("Department: "+this.department);
      console.log("Bonus: "+(this.calculateBonus()));
   }
}
let e1 = new Employee("James","Developer",50000);
let m1 = new Manager("John","Manager",100000,"Accounts",5);
e1.getDetails();
m1.getManagerDetails();
```

Output:-

```
Name: James
Position: Developer
Salary: 50000
Name: John
Position: Manager
Salary: 100000
Department: Accounts
Bonus: 105000
```

5. **Construct an array of 10 numbers and perform the following operations with higher-order array methods.**

   a. Display all the numbers with the help of forEach loop

   b. Multiply each number with 100 and construct a new array. Display the newly constructed array.

   c. Construct a new array from the existing array which consists of the numbers which are divisible by both 3 and 5.

   d. Remove a number from the array with the help of filter function

e. Find and display the sum of all the numbers of the array.

f. Sort the array in ascending order and display

Code:-

```
//Sunil Kumar Sahoo, 23MMCI38, 16
let arr=[18,20,22,37,1,3,8,9,7,12,15];
 arr.forEach((n)=>{
    console.log(n);
 })
 console.log("After multiplying with 100");
 arr.forEach((n)=>{
    let new_arr=n*100;
    console.log(new_arr);
 })
 let new_arr1=arr.map((n)=>{
    if(n%3==0 && n%5==0){
      return n;
    }
 })
 console.log(new_arr1);
 let remove_num=arr.filter((n)=>{
    if(n==20){
       return false;
    }
    else{
       return true;
    }
 })
 console.log(remove_num);
 let sum_arr=arr.reduce((sum,n)=>{
    return sum+n;
 },0);
console.log(sum_arr);
arr.sort((a,b)=>a-b);
console.log(arr);
```

Output:-

```
18
20
22
37
1
3
8
9
7
12
15
After multiplying with 100
1800
```

```
2000
2200
3700
100
300
800
900
700
1200
1500
[15]
152
[ 1, 3, 7, 8 , 9, 12, 15, 18, 20, 22, 37]
```

6. **Consider the given array of objects and perform the operations using higher- order array methods only.**
employees=[{id:1,name:'John Doe', age:28,salary:50000, skill:['JavaScript', 'React']},
{id:2,name:'Jane Smith', age:35,salary:60000, skill:['JavaScript', 'Node.js', 'MongoDB']},
{id:3,name:'Alice Johnson', age:32,salary:55000, skill: ['HTML', 'CSS'] },
{id:4,name: 'Bob Williams', age:62,salary: 70000, skill:['Java', 'Spring'] },
{id:5,name: 'Eva Davis', age: 28, salary:52000, skill: ['Python', 'Django']},];
{id:6,name:'Mike Brown', age:62,salary:75000, skill:['JavaScript', 'TypeScript']},
{id:7,name: 'Sara Miller', age:30, salary:58000, skill: ['SQL', 'C#']}
i. For every employee construct and display a string like "John Doe is 20years old and getting a salary of Rs 45000"
ii. Construct an array consisting of only the names of all the employees and display the names
iii. Find the average salary of all the employee
iv. Find the employee names who knows JavaScript
v. Remove the employee with the employee id 3
vi. Remove the employee from the array who is more than 60 years old

Code:-

```
//Sunil Kumar Sahoo, 23MMCI38, 16
employees = [
   { id: 1, name: 'John Doe', age: 28, salary: 50000, skill: ['JavaScript', 'React'] },
   { id: 2, name: 'Jane Smith', age: 35, salary: 60000, skill: ['JavaScript', 'Node.js', 'MongoDB']
},
   { id: 3, name: 'Alice Johnson', age: 32, salary: 55000, skill: ['HTML', 'CSS'] },
   { id: 4, name: 'Bob Williams', age: 62, salary: 70000, skill: ['Java', 'Spring'] },
   { id: 5, name: 'Eva Davis', age: 28, salary: 52000, skill: ['Python', 'Django'] },
   { id: 6, name: 'Mike Brown', age: 62, salary: 75000, skill: ['JavaScript', 'TypeScript'] },
   { id: 7, name: 'Sara Miller', age: 30, salary: 58000, skill: ['SQL', 'C#'] }
   ];
  employees.forEach(emp => {
    console.log(`${emp.name} is ${emp.age} years old and getting a salary of Rs
${emp.salary}`);
 });
let names=employees.map(n=>n.name);
names.forEach(n=>console.log(n));
```

```
let total_sal=employees.reduce((sum,s)=>sum+s.salary,0)
console.log(total_sal/employees.length);
let count=0;
let skill_known=employees.filter((n)=>{
   return n.skill.includes("JavaScript")
 })
 skill_known.forEach(n=>console.log(n.name));
let remove_id=employees.filter((n)=>{
    if(n.id==3){
       return false;
   }
   else{
      return true;
   }
 })
 console.log(remove_id);
let remove_age=employees.filter((n)=>{
   if(n.age>60){
      return false;
   }
   else{
      return true;
   }
})
console.log(remove_age);
```

Output:-

```
John Doe is 28 years old and getting a salary of Rs 50000
Jane Smith is 35 years old and getting a salary of Rs 60000
Alice Johnson is 32 years old and getting a salary of Rs 55000
Bob Williams is 62 years old and getting a salary of Rs 70000
Eva Davis is 28 years old and getting a salary of Rs 52000
Mike Brown is 62 years old and getting a salary of Rs 75000
Sara Miller is 30 years old and getting a salary of Rs 58000
John Doe
Jane Smith
Alice Johnson
Bob Williams
Eva Davis
Mike Brown
Sara Miller
60000
John Doe
Jane Smith
Mike Brown
[
  {
    id: 1,
    name: 'John Doe',
    age: 28,
    salary: 50000,
    skill: [ 'JavaScript', 'React' ]
  },
  {
    id: 2,
    name: 'Jane Smith',
    age: 35,
    salary: 60000,
    skill: [ 'JavaScript', 'Node.js', 'MongoDB' ]
  },
  {
    id: 4,
    name: 'Bob Williams',
    age: 62,
    salary: 70000,
    skill: [ 'Java', 'Spring' ]
  },
  {
    id: 5,
    name: 'Eva Davis',
    age: 28,
    salary: 52000,
    skill: [ 'Python', 'Django' ]
  },
  {
    id: 6,
    name: 'Mike Brown',
    age: 62,
    salary: 75000,
    skill: [ 'JavaScript', 'TypeScript' ]
  },
  {
    id: 7,
    name: 'Sara Miller',
    age: 30,
    salary: 58000,
    skill: [ 'SQL', 'C#' ]
  }
]
[
  {
    id: 1,
    name: 'John Doe',
    age: 28,
    salary: 50000,
    skill: [ 'JavaScript', 'React' ]
  },
 {
   id: 2,
   name: 'Jane Smith',
   age: 35,
   salary: 60000,
   skill: [ 'JavaScript', 'Node.js', 'MongoDB' ]
 },
 {
   id: 3,
   name: 'Alice Johnson',
   age: 32,
   salary: 55000,
   skill: [ 'HTML', 'CSS' ]
 },
 {
   id: 5,
   name: 'Eva Davis',
   age: 28,
   salary: 52000,
   skill: [ 'Python', 'Django' ]
 },
 {
   id: 7,
   name: 'Sara Miller',
   age: 30,
   salary: 58000,
   skill: [ 'SQL', 'C#' ]
 }
]
```

7. **With the help of asynchronous functions of NodeJS „fs" module perform the file read, write, append, rename and delete operations.**

Code:-

```
//Sunil Kumar Sahoo, 23MMCI38, 16
const fs=require('fs')
 fs.writeFile("exp.txt","Hello!",(err)=>{
    if(err){
        consolelog(err);
    }
    else{
        console.log("written");
    }
 })
 fs.readFile("exp.txt",'utf8',(err,data)=>{
    if(err){
        console.log(err);
    }
    else{
        console.log(data)
    }
})
fs.appendFile("exp.txt","\n good morning",(err)=>{
    if(err){
        consolelog(err);
    }
    else{
        console.log("appended");
    }
})
fs.rename("exp.txt","newname.txt",(err)=>{
    if(err){
        console.log(err);
    }
    else{
        console.log("renamed")
    }
})
fs.unlink("newname.txt",(err)=>{
    if(err){
        console.log(err);
    }
    else{
        console.log("removed")
    }
})
```

Name: Sunil Kumar Sahoo
SIC: 23MMCI38
Roll: 16
Group: A2