# Department of Computing

**ALINA SHOAIB QURESHI**

**BESE-9A**

**246312**

**SE-312: Software Construction**

**Class: BESE 9AB**

# Lab 03: Intro to ES6

**Date: 8th March 2021**

**Time: 09:00-11:50pm & 02:00-04:50pm**

# Instructor: Dr. Seema Jehan

**Lab Engineer: Mr. Aftab Farooq**

# 

# Lab 03: Intro to ES6

**Objectives**

The objective of this lab is helping students to familiarize themselves with basic concepts of the ES6 constructs. They will practice the concept of classes, subclasses, template strings, default parameters, maps, arrow functions and destructuring.

**Tools/Software Requirement**

Notepad, browser

**Helping Material:**

File Uploaded on LMS

**Lab Tasks:**

**TASK1:**

Suppose that you're working in a small town administration, and you're in charge of two town elements:

1. Parks

2. Streets

It's a very small town, so right now there are only 3 parks and 4 streets. All parks and streets have a name and a build year.

At an end-of-year meeting, your boss wants a final report with the following:

1. Tree density of each park in the town (formula: number of trees/park area)

2. Average age of each town's park (formula: sum of all ages/number of parks)

3. The name of the park that has more than 1000 trees

4. Total and average length of the town's streets

5. Size classification of all streets: tiny/small/normal/big/huge. If the size is unknown, the default is normal

All the report data should be printed to the console.

HINT: Use some of the ES6 features: classes, subclasses, template strings, default parameters, maps, arrow functions, destructuring, etc.

|  |
| --- |
| Solution |
| Task 1 Code:  class GeneralElement {    constructor(name, buildYear) {      this.name = name;      this.buildYear = buildYear;    }  }  class Park extends GeneralElement {    constructor(name, buildYear, area, numTrees) {      super(name, buildYear);      this.area = area; //km2      this.numTrees = numTrees;    }      //method treeDensity()    treeDensity() {      const density = this.numTrees / this.area;      console.log(        `${this.name} has a tree density of ${density} trees per square km.`      );    }  }  class Street extends GeneralElement {    constructor(name, buildYear, length, size) {      super(name, buildYear);      this.length = length;      this.size = size;    }      //method classifyStreet    classifyStreet() {      const classification = new Map();      classification.set(1, "tiny");      classification.set(2, "small");      classification.set(3, "normal");      classification.set(4, "big");      classification.set(5, "huge");      console.log(        `${this.name}, build in ${this.buildYear}, is a ${classification.get(this.size)} street.`      );    }  }  function calc(arr) {    const sum = arr.reduce((prev, cur, index) => prev + cur, 0);    return [sum, sum / arr.length];  }  function reportParks(p) {    console.log("----PARKS REPORTS----");    // Density    p.forEach((el) => el.treeDensity());    // Average age    const ages = p.map((el) => new Date().getFullYear() - el.buildYear);    const [totalAge, avgAge] = calc(ages);    console.log(`Our ${p.length} parks have an average of ${avgAge} years.`);    // Which park has more than 1000 trees;    const i = p.map((el) => el.numTrees).findIndex((el) => el >= 1000);    console.log(`${p[i].name} has more than 1000 trees.`);  }  function reportStreets(s) {    console.log("---- STREETS REPORT ----");    //Total and average length of the town's streets    const [totalLength, avgLength] = calc(s.map((el) => el.length));    console.log(      `Our ${s.length} streets have a total length of ${totalLength} km, with an average of ${avgLength} km.`    );    //Classify sizes    s.forEach((el) => el.classifyStreet());  }  const allParks = [    new Park("Green Park", 1997, 0.2, 215),    new Park("Askari Park", 1964, 2.9, 3541),    new Park("F10 Park", 1793, 0.4, 949),  ];  const allStreets = [    new Street("Ocean Avenue", 2009, 1.1, 4),    new Street("Allama Iqbal Street", 2015, 2.7, 2),    new Street("10th Street", 2000, 0.8),    new Street("Sunset Boulevard", 1992, 2.5, 5),  ];  reportParks(allParks);  reportStreets(allStreets);  Task 1 Output Screenshot: |

### Deliverables

Compile a single word document by filling in the solution part and submit this Word file on LMS. This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva related to the tasks. The viva has a weightage of 5 marks. Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems with submissions on LMS, submit your Lab assignments by emailing it to Mr. Aftab Farooq: [aftab.farooq@seecs.edu.pk](mailto:aftab.farooq@seecs.edu.pk).