**WORKSHOP 2**

Due Date: Thursday June 6, 11.59 PM

*This workshop must be completed individually without any outside collaboration. All*

*work must be your own. Copying or reproducing the work done by others (in part or*

*in full) or letting others to copy or reproduce your own work is subject to significant*

*grade reduction or getting no grade at all and/or being treated as academic*

*dishonesty under the College’s Academic Dishonesty Policy.*

*• Your application must compile and run upon download to receive any mark.*

*• To submit the workshop, please follow the Submission Guideline provided at the end*

*of this document.*

*• You must submit your workshop by the due date. Late submissions policy is specified*

*in the Academic Procedures for Evaluations document available through the class*

*plan on Blackboard.*

**Description:**

The second workshop lets the student practice about designing a bit more complicated

JavaFx screen. Create an Online Job Application portal which helps you to practice

different design layouts and model classes.

**Task:**

So you are tasked to design a job portal which the company will use to hire new employees. Your job is to design a desktop application, which takes in the applicant information and stored it in a data structure. An applicant can provide their name, phone number, and add their skills to a list.

Requirements:

· Following the MVC design pattern you are required to create model classes in this

project which will deal with multiple model classes like Applicant and Company at

the minimum.

1. **Applicant Class**:

• This class represents an individual job applicant.

• Attributes include the applicant's name, phone number, and any other

applicant-specific details.

• Include a collection or list of selected companies as a reference. Each

selected company represents the companies the applicant is interested in.

2. **Job Role Class**:

• This class represents Job role the applicant is applying to.

• Attributes may include the jol role, description, and other things which are specific to the job role.

· You are required to use Scene Builder to design the Job Posting application a

snapshot which will be discussed in the lab session.

**EVENT HANDLERS**

1. **Submit Application Button**:

• Event: When the applicant clicks the "Submit Application" button.

• Functionality: This event handler should gather the applicant's

information (name, phone number) and the job role applied. It

then creates an instance of the **Applicant** class, populates it with this

information, and stores it in the data structure (e.g., ArrayList).

2. **View Job Details Button**:

• Event: When the applicant clicks a "View Details" button associated

with a specific Job Role.

• Functionality: This event handler should display detailed information

about the selected Job Role, such as the description,

required skills, etc. It can open a new window or display the information within the same scene.

3. **Clear Button**:

• Event: When the applicant clicks the "Clear" button.

• Functionality: This event handler should clear all the fields on the

application form, allowing the applicant to start fresh.

4. **Exit Button**:

• Event: When the applicant clicks the "Exit" button.

• Functionality: This event handler should close the application.

•

5.**Skills Text Input**:

• Event: When the applicant enters their skills in the text box.

• Functionality: This event handler should capture the skills entered by the

applicant in real-time as they type or when they finish entering skills.

You can store this information as a list or a string in the **Applicant** class

to keep track of the skills the applicant possesses.*Workshop Header*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*Workshop #*

*Course:<subject type> - Semester*

*Last Name:<student last name>*

*First Name:<student first name>*

*ID:<student ID>*

*Section:<section name>*

*This assignment represents my own work in accordance with Seneca Academic*

*Policy.*

*Signature*

*Date:<submission date>*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*Code Submission Criteria:*

*Please note that you should have:*

*· Appropriate indentation.*

*· Proper file structure*

*· Follow java naming convention*

*· Do Not have any debug/ useless code and/ or files in the assignment*

*Deliverables and Important Notes:*

*All these deliverables are supposed to be uploaded on the blackboard once done.*

*· Design should be discussed/ created during the lab. 25%.· Complete the code behind as the part of your DIY. 50%*

*· Submit a reflect.txt file with the submission. 10%*

*Questions to be answered for the reflection:*

*· Class Design*

*o Why you have single class or multiple classes in your design.*

*· Data structure*

*o Why choosing your particular data structure (used in the solution).*

*o Compare your chosen data structure with at least one more data structure.*

*§ Your comparison should be talking about performance.*

*· Video submission explaining core code pointers and showing the full working*

*application (3 – 8 minutes max). 15%*

*· All submission goes to Black Board.*

*· Your submission should include*

*o Video file with audio*

*o Reflect.txt file*

*o Complete zipped project.*

*· Late submissions would result in additional 10% penalties for each day or part of it.*

*Remember that you are encouraged to talk to each other, to the instructor, or to*

*anyone else about any of the workshops, but the final solution may not be copied from*

*any source.*