**Project Assessment**

**GENERAL INSTRUCTIONS:** Please carefully read the below instructions

The objective of this assessment is to check your ability to complete a project as per the provided “Project Design”.

**You are expected to –**

1.      Write the source code for the classes, methods and packages **EXACTLY** as mentioned in the “**Project Design**” section.

2.      Ensure that the names of the packages, classes, methods and variables **EXACTLY MATCH** with the names specified in the “Project Design” section.

3.      Understand the project requirements and ACCORDINGLY WRITE the code and logic in the classes and methods so as to meet all given requirements.

**Creating the project and testing it –**

1.      You are expected to create your project locally using eclipse (or any other IDE) on your desktop.

2.      Once you are ready with the code, you should upload the src folder of your project in .zip format, using the “Upload Zip File” button.

IMPORTANT NOTE 1 : The extension of the zip file should be ONLY .zip (any other zip formats such as .7z  will produce unexpected results)

IMPORTANT NOTE 2 : The .zip file should contain zip of ONLY the src folder structure from your project. (If the zip file has anything other than the src folder structure, the result will be unexpected. Do not zip the entire project folder structure. Just do the zip of the src folder structure and upload it)

IMPORTANT NOTE 3 : The name of the .zip file should be <your employee number>.zip For e.g., if your emp no. is 12345, the zip file should be named 12345.zip.

3.      After uploading the zip file, you can click on “Compile & Test” button and the assessment engine will compile your source code and test it using its pre-defined test-cases.

4.      If some of the test-cases fail, you can make the fixes in your source code locally on your desktop, and again repeat the above two steps.

5.      Once you are finished with all the fixes, you can click on “Final Submission” button, which will show you the final result/score.

**NOTE that –**

6.      The assessment engine will create objects and invoke methods as per the project design, and while doing so, it will use your packages, classes and methods. If your packages, classes and methods have a name mismatch or method prototype mismatch w.r.t the expected “Project Design”, the tool will show it as an ERROR. If your packages, classes and methods match as per the names but do not perform the expected functionality, the tool will show it as a FAILURE.

7.      Unless specified in the Project Design, DO NOT use **System.exit(0)** anywhere in your code. Using **System.exit(0)** in your project code will cause the CPC test engine to exit and it will not be able to run all test-cases.

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

**Interview Process System**

**Project Objective:**

Create a console based Java application that would allow the clerk of an organization to scrutinize the profiles of the candidates who apply for a job to their organization.

There are two types of candidates who apply for the Job.

1.      Internal Reference – Profiles who are referred by people who are already working in the organization

2.      External Reference – Profiles who are recommended through some agency

**Process Flow:**

1.      In the main method create a ProfileEvaluator object

2.      Initialize it with values with the requirements of the job

3.      Call CreateObject method of Profile Evaluator class, passing the details of the candidate and receive the respective applicant object

4.      Call the validateObject method of ProfileEvaluator and validate the Profile

5.      If the Profile is valid, call the ProcessProfile to see if the Applicant is selected or not

**Project Design:**

**A. System Design:**

|  |  |
| --- | --- |
| **Name of the package** | **Usage** |
| com.wipro.ips.entity | This package will contain the candidate profile related classes |
| com.wipro.ips.exception | This package will contain the user defined exception class |
| com.wipro.ips.main | This package will contain the MainClass that is used to test the application |
| com.wipro.ips.service | This package will contain the class that is used to validate the data and process the profile for given role and identify the process of interview |

**Package: com.wipro.ips.exception**

|  |  |  |
| --- | --- | --- |
| **Class** | **Method and Variables** | **Description** |
| **DataValidationException** |  | **An Exception Class** |
|  | public String toString() | Returns the message “**INVALID DATA”** |

**Package: com.wipro.ips.entity**

|  |  |  |
| --- | --- | --- |
| **Class** | **Method and Variables** | **Description** |
| **Applicant** |  | **Abstract Class** |
|  | private int applicantId |  |
|  | private String applicantName | Not null |
|  | private int yrsOfExp |  |
|  | private String skills | Skills value would be given as different skills separated by (**,**) eg: “java,sql,html” |
|  | private double expectedSalary |  |
|  | Getter and Setter methods for the above fields |  |
|  | public Applicant(int applicantId,String applicantName,int yrsOfExp, String skills, double expSalary) | **A parameterized constructor** used to initialize the applicant object |

**Package: com.wipro.ips.entity**

|  |  |  |
| --- | --- | --- |
| **Class** | **Method and Variables** | **Description** |
| ExternalReferral |  | Inherits Applicant Class |
|  | public ExternalReferral (int applicantId,String applicantName,int yrsOfExp, String skills, double expSalary, String agencyName) | **A parameterized constructor** used to initialize the  applicant object |
|  | String agencyName |  |

**Package: com.wipro.ips.entity**

|  |  |  |
| --- | --- | --- |
| **Class** | **Method and Variables** | **Description** |
| InternalReferral |  | Inherits Applicant Class |
|  | public InternalReferral (int applicantId,String applicantName,int yrsOfExp, String skills, double expSalary, int employeeId) | **A parameterized constructor** used to initialize the applicant object |
|  | int employeeId |  |

**Package: com.wipro.ips.service**

|  |  |  |
| --- | --- | --- |
| **Class** | **Method and Variables** | **Description** |
| **ProfileEvaluator** |  | Class |
|  | private int JobId |  |
|  | private String JobName |  |
|  | private String requiredSkill | requiredSkills value would be given as different skills separated by (**,**) eg: “java,sql,html” |
|  | private int expMinExperience |  |
|  | private int expMaxExperience |  |
|  | Generate Getters and Setters for the above variables |  |
|  | public Applicant createObject(char referralType, int applicantId, String applicantName, int yrsOfExp, String skills, double expSalary,  String referralInfo) | This method is used to create the right type of applicant object and return the same.  For example  If referralType is ‘**I**’ create InternalReferral object where referralInfo is assumed as empid. Return the created InternalReferral object  Else if referralType is ‘**E**’ create ExternalReferral object where referralInfo is assumed as agencyName  Return the created ExternalReferral object  If referral Type is neither ‘E’ nor ‘I’ return null.  **Note:** empid of internal referral is int, given referralInfo String should be converted to integer and set. |
|  | public String validateObject(Applicant applicant) throws DataValidationException | This method checks the validation of the applicant data.           Applicant Object should be not null           Applicant Name cannot be blank or null           In case if any of the above 2 conditions fails, the function should throw **DataValidationException**           If Years of total experience is <= 0 return “**INVALID EXPERIENCE**”           If Expected Salary is <=0 return “**INVALID SALARY**”           When all conditions succeed return “**VALID**”.  **Note:** do not handle the exception in the method |
|  | public String processProfile(Applicant applicant) | This method is used to check whether the applicant is suitable for the mentioned job requirement    Compare the applicant’s given experience and skills with the Job Opening’s expected min experience, expected max experience and expected skills in the ProfileEvaluator class variables.  The following are the conditions of a valid data           The Applicant’s experience should be between reqMinExp and reqMaxExp. If not return “**NON ELIGIBLE EXPERIENCE**”           Return “**NON ELIGIBLE SKILLS**” If none of the  reqSkills matches applicant’s skills           If all the above condition passes return “**SELECTED**” |

**Package : com.wipro.ips.main**

|  |  |  |
| --- | --- | --- |
| **Class** | **Method and Variables** | **Description** |
| **MainClass** |  | **Main Class** |
|  | public static void main(String[] args) | Creates ProfileEvaluator object and set the following data  JobId, JobName, requiredSkill, expMinExperience and expMaxExperience .    Get Applicant data from the user and after receiving all this data, invoke the createObject methodof **ProfileEvaluator** class  followed by validateObject method of the same class finally match the profile using the processProfile method and test your program. Sample main program code given below |

**The main method of the Main Class may look like this:**

**public** **static** **void** main(String[] args)

{

              ProfileEvaluator pe = **new** ProfileEvaluator();

              pe.setJobId(101);

              pe.setJobName("Java Developer");

              pe.setExpMinExperience(2);

              pe.setExpMaxExperience(5);

              pe.setRequiredSkill("java,sql");

**int** applicantId = 1001;

              String applicantName = "abc";

**int** yrsOfExp = 2;

              String skills = "dos,oracle,sql,java,servlets,jsp,spring";

**double** expSalary = 120000;

**char** referralType = 'I';

              String referralInfo = "12345";

              Applicant applicant = pe.createObject(referralType, applicantId, applicantName, yrsOfExp, skills, expSalary, referralInfo);

**try**

              {

                     String status = pe.validateObject(applicant);

**if**(status.equals("VALID"))

                     {

                           String result = pe.processProfile(applicant);

                           System.*out*.println("Process Result is :"+result);

                     }

              }

**catch** (DataValidationException e) {

                     e.printStackTrace();

              }

       }

Change the values of variables declared in the main method and test your program.

Test Cases:

1.       Test for Applicant creation based on referralType through createObject

2.       Test for Applicant creation by Invalid or wrong referralType through createObject

3.       Test for Null Applicant object through validateObject

4.       Test for Invalid Applicant Name through validateObject

5.       Test for Invalid Expected Salary through validateObject

6.       Test for Invalid Experience through validateObject

7.       Test for Valid Applicant object through validateObject

8.       Test for Non Eligible Experience through processProfile

9.       Test for Non Eligible Skills through processProfile

10.   Test for Eligible Candidate through processProfile