L2-Project Assignment

TopGear
Engage | Be Future Ready

Project Title – Java Programming Essentials –L1-Assignment

Proejct Type	Project Name/ID	Hosted By TG/Acccount	Duration	Proejct Owner /Account
L1 Assignment	Java Programming Essentials –L1-Assignment	TopGear	5 PDs	NA

Table of Contents

About This Project

This is L1 level training on advanced topics of Java programming language.

Who Is Eligible

All users who have either completed TopGear Core Java L1 training or experienced with OOPS and essential Core Java programming.

Pre-Requisites

1. User need to have experience in Object Oriented Programming Concepts (OOPS) and Core Java.
2. Should have completed Core Java L1 level training.
3. User is recommended to consult a TopGear mentor and clarify doubts/queries (if any) before joining the project.
4. Project to be submitted to TopGear project repository using GIT with appropriate documentation and packaged as guided by TopGear mentor.
5. Irrelevant submission will be barred from review and will be considered not completed which will be re-opened with appropriate comments.

Scope of Work & Deliverables

1. Coding & Testing
2. Source files packaged as guided.
3. To understand executing and submission process of TopGear Training and Case study projects user can refer following documents on the portal:

https://topgear-app.wipro.com/sites/default/files/steps_to_execute_topgear_training_case_study_projects_v-1-2-1.pdf

Note: User is recommended to read this document thoroughly before starting with their respective TopGear project.

Technologies

This assignment uses Java language to explore and learn advanced topics of programming.

Training Tasks

1. Task to implement exception handling for given scenario:

Sometimes it is required to develop meaningful exceptions based on application requirements. We can create our own exceptions by extending 'Exception' class.
Consider Class Student
Class Student
Attributes:
Name, Roll No, class, Array [lang1Marks, lang2Marks, lang3Marks, subj1Marks, subj2Marks, subj3Marks], result

Methods
Getters and setters for all, the method set result must throw and incorrect exception if the avg of all marks is below 50 or if any individual marks is below 50

2. Task to implement Java Threads by for given scenario:

Imagine the following scenario. You are preparing for tomorrow's final examination and feel a little hungry. So, you give your younger brother ten bucks and ask him to buy a pizza for you. In this case, you are the main thread and your brother is a child thread. Once your order is given, both you and your brother are doing their job concurrently (i.e., studying and buying a pizza). Now, we have two cases to consider. First, your brother brings your pizza back and terminates while you are studying. In this case, you can stop studying and enjoy the pizza. Second, you finish your study early and sleep (i.e., your assigned job for today - study for tomorrow's final exam - is done) before the pizza is available. Of course, you cannot sleep; otherwise, you won't have a chance to eat the pizza. What you are going to do is to wait until your brother brings the pizza back.
A thread can execute a thread join to wait until the other thread terminates
A parent thread may join with many child threads created by the parent. Or, a parent only join with some of its child threads, and ignore other child threads. In this case, those child threads that are ignored by the parent will be terminated when the parent terminates.

The Skeleton is described as follows

```
import java.util.ArrayList;
import java.util.List;

public class MyThreadJoin {

    public static List<String> names = new ArrayList<String>();

    public static void main(String a[]){
//your code goes here
    }
}

class SampleThread extends Thread{

    public void run(){

         }
        }
}
```

3. Task to implement HTML tag remover Java utility :

In case if a string contains html tags, develop a class which helps to trim the html tags from the string. The class must use regular expression to trim the html tags from the string.

Example Input / Output

<B>I don't want this to be bold<\B>
I don't want this to be bold

```
class MyHtmlTagRemover {

    public static String replace(String a[]){
        //place your code
    }
}
```

4. Task to implement given class relationship and related functionalities:

We have two classes called CompAEmp and CompBEmp extending Emp class. We have a generic class called MyEmployeeUtil, where we have utilities to perform employee functions irrespective of which company emp belongs too. This class accepts subclasses of Emp. Incase if we want to compare salaries of two employees, how can we do using MyEmployeeUtil class?

```
public class MyWildcardEx {

    public static void main(String a[]){

        MyEmployeeUtil<CompAEmp> empA
                    = new MyEmployeeUtil<CompAEmp>(new CompAEmp("Ram", 2000
0));
        MyEmployeeUtil<CompBEmp> empB
                    = new MyEmployeeUtil<CompBEmp>(new CompBEmp("Krish", 30
000));
        MyEmployeeUtil<CompAEmp> empC
                    = new MyEmployeeUtil<CompAEmp>(new CompAEmp("Nagesh", 2
0000));
        System.out.println("Is salary same? "+empA.isSalaryEqual(empB));
        System.out.println("Is salary same? "+empA.isSalaryEqual(empC));
    }
}

class MyEmployeeUtil<T extends Emp>{



    //// create some utility methods to do employee functionalities
    //
    //
    //
}
```

```
class Emp{

    private String name;
    private int salary;

    public Emp(String name, int sal){
        this.name = name;
        this.salary = sal;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getSalary() {
        return salary;
    }
    public void setSalary(int salary) {
        this.salary = salary;
    }
}

class CompAEmp extends Emp{

    public CompAEmp(String nm, int sal){
        super(nm, sal);
    }
}

class CompBEmp extends Emp{

    public CompBEmp(String nm, int sal){
        super(nm, sal);
    }
}
```

5. **Task to learn compressing and un-compressing file(text) using java.util.zip.GZIPOutputStream API:**

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.zip.GZIPOutputStream;
```

```
public class MyFileGZipExamp {

    public void doGzip(String filePath){

    }
      public void doUnGzipFile(String filePath) {



    }
    public static void main(String a[]){

        MyFileGZipExamp mfg = new MyFileGZipExamp();
        mfg.doGzip("C:/test.txt");

        mfg.doUnGzipFile("C:/myGzip.gzip");
    }
}
```

6. Task to learn formatting a String using Java Regular Expressions:

Match a string pattern with a regular expression. String.matches () method helps to match the string with a regex. Identify if the passed argument has "www' in the parameter.

```
public class MyStrMatches {



    public static void main(String a[]){



        }

    }

}
```

Environment

To execute this training project user can use own desktop/laptop set with essential Java development environment (need GitBash tool) or can use TopGear provided VDI (Spring SOA Hibernate) which is configured for Java and Java EE development

Completion Criteria

- Employee should complete the project within timeline provided.
- In cased user has not follows appropriate steps or job is partly finished, then project can be reopened/reverted back.
- Employee must follow all rules and task as instructed to avoid rejection/reopening of projects.
- Once the project is closed successfully after submission and review, user will earn # reward points and the user will visible to appropriates practices and accounts.
- If the user has completed a Use case, project or Solution project, he/she must update his/her Wipro profile add the project details he worked.
- User should update efforts against TopGear projects in time sheet.

Assumptions

Users joining this program have knowledge/understanding of OOPS concepts and Core Java programming APIs.

Project Submission Guidelines

- Login https://topgear.wipro.com
- Go to Java-J2EE → Java→Training Tab
- Join Java Programming Essentials -L1-Assignment project.
- Practice on assignment as per project specification.
- Once complete the project work, submit/check-in the same as instructed to TopGear GitLab.
- User can refer following document to understand execution and submission procedure of TopGear training and case study projects

https://topgear-app.wipro.com/sites/default/files/steps_to_execute_topgear_training_case_study_projects_v-1-2-1.pdf

References

- http://beginnersbook.com/2013/04/java-exception-handling/
- http://www.javatpoint.com/multithreading-in-java
- https://docs.oracle.com/javase/7/docs/api/java/lang/String.html
- https://docs.oracle.com/javase/tutorial/java/generics/
- http://www.journaldev.com/966/java-gzip-example-compress-decompress-file
- http://www.vogella.com/tutorials/JavaRegularExpressions/article.html