# **Abstract - Lab**

1. Create a new workspace => file  =>switch workspace => other => add at the end /claim-workspace.  
  
2. Create a new project, name it AbstractLab.

3. Create a new abstract class called Person. File  => new => class, then add the word abstract after the modifier. Eg: public abstract class Person{    }

4. Add an abstract method called taxes.

To do this we add the word abstract before the method modifier. The method should return a double and should take a double as a parameter. Add protected fields : first name and last name class variables

Eg: public abstract double taxes(double amount);

5. Create two other classes called Student and Employee. Both should extend the Person class.

6 override the taxes method from the parent class.

7. If you are a student you should pay 2% in tax..

8. If you are Employed you should pay 20% in taxes .

Eg: public double taxes(double amount){

return amount \* .02;

//Please note this is only for calculating taxes for a student object

}

9. Create a method that's called calculate telephone bill and calculate all other bills.

 These methods should return a double and take a double as a parameter

 Add these methods to the Person Class

 Eg: public double calculateTelePhoneBill(double amount){

 return this.taxes(amount) + amount;

       }

10. Create a class called Application.

11. Add the main method to this class.

12. Create a new Student object;  eg: Student s1 = new Student();

13. Create a new Employee object

 14. Make a call to the student object calculate telephone bill method

eg: double stuTelPhoneBill = s1.calculateTelePhoneBill(120.5);

15. Print this data

16. Repeat this for the Employee object

17. Create a new Student object. Eg: Person p1 = new Student();

18. Create a new Employee object. Eg: Person p2 = new Employee();

19. Make calls to the Person objects to print telephone bill;

20. Create toString method for the Student object and the Person Object;

21. Set the user name for all 4 objects.

22. Print this data.

22.a Make a call to the tax method in the person objects and print the data

23. Create an interface called Bills. File => new =>interface.

24. Add abstract methods gas bill and car insurance.

25. Implement this interface.  To do this, go to Student and Employee class. After the extends

      Person add implements Bills.

26. Create another class and name it school.

27. Add an arrayList to store all the student objects

 Eg List<Student> students = new ArrayList<Student>():

 List => this represent the List class interface and ArrayList is the implementation of this

 Interface. The <Student> represent that we want to store a List of Student objects.

28. Add a second arrayList to store all the Employees.

29. Add all the student and employee objects to the school array.

 => in the main method create an arrayList and add all the student objects to this list

 => eg : studentArray.add(s1);

 => add this array list to the School object  => repeat this for the Employee object