# Beyond The Basics

**Grade settings**: Maximum grade: 100
**Disable external file upload, paste and drop external content**: Yes
**Based on**: Beyond The Basics
**Run**: Yes **Evaluate**: Yes
**Automatic grade**: Yes

Beyond The Basics is one of the well-known Insurance companies, that wanted to automate the process of calculating the maturity amount of each policy, which will be easily accessed by their policyholders through their websites. As a developer help the company to implement the requirements.

### Component Specification: Insurance

| Type(Class) | Attributes | Methods |
|---|---|---|
| **Insurance** | String policyNumber<br>String insuranceProvider<br>double coverageAmount<br>String coverageType<br>int durationInYears | Necessary getters, setters and five argument constructors are provided as part of the code skeleton. |

### Functional Requirement 1: Extract the details of Insurance and create an object of the Insurance class.

| Type(Class) | Method | Responsibilities |
|---|---|---|
| **UserInterface** | public static Insurance **extractDetails**(String insuranceDetails) | This method should accept **insuranceDetails** as an argument and extracts the properties of **Insurance** from the argument and set these values to the **Insurance** object. |

### Functional Requirement 2: Calculate the maturity amount for the given duration.

| Type(Class) | Method | Responsibilities |
|---|---|---|
| **Insurance** | public double **calculateMaturityAmount**() | This method should calculate the maturity amount for the given duration in years and return the result. |

|  |  |  |
|---|---|---|
|  |  | If the durationInYears greater than zero then, calculate the maturityAmount.<br>***Condition:***<br>- ***coverageAmount should be positive else return -1.***<br>- ***If durationsInYears Less than or equal to zero then, return -1.*** |

**Formula to calculate maturity amount,**

**MaturityAmount=coverageAmount**

**maturityAmount += (maturityAmount * 0.05) ^durationInYear.**

**' ^ ' symbol represents power.**

Let, coverageAmount=1000, durationInYears=2

For 1st year,

MaturityAmount=1000+(1000*0.05)=1050

For 2nd year,

MaturityAmount=1050+(1050*0.05)=1102.50

**The main method in the UserInterface class is excluded from evaluation. You are free to write your own code in the main method to invoke the business methods to check its correctness.**

**Note:**

- In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.
- Ensure to follow the object-oriented specifications provided in the question
- Ensure to provide the names for classes, attributes, and methods as specified in the question.
- Adhere to the code template, if provided.

**Sample Input 1**

Enter the details

**STR45789:Star Insurance:1000:Full:2**

**Sample Output 1**

Policy Number: STR45789

Insurance Provider: Star Insurance

Coverage Amount: 1000.0

Coverage Type: Full

Duration in years: 2

Maturity Amount: 1102.50

**Sample Input 2**

Enter the details

**ICIC102289:ICICI Insurance:10000:Half:-5**

**Sample Output 2**

Invalid details

**Explanation :** As durationInYears is invalid.

**Sample Input 2**

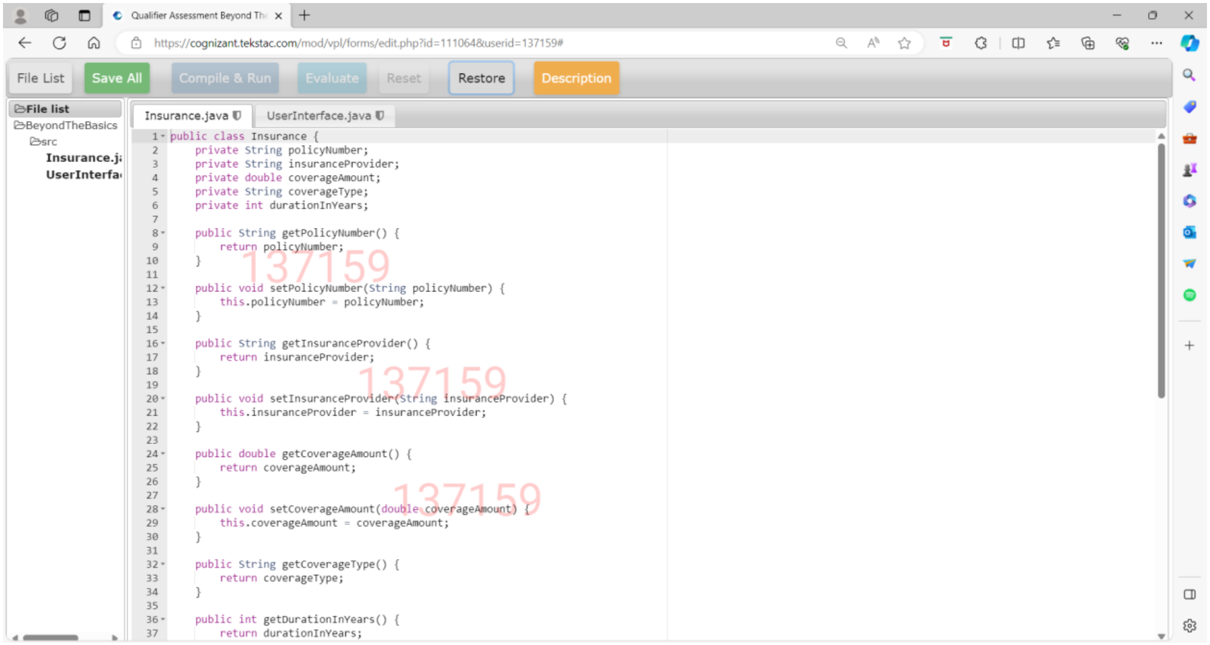
Enter the details

**ICIC132712:Prudential Insurance:-10000:Half:5**

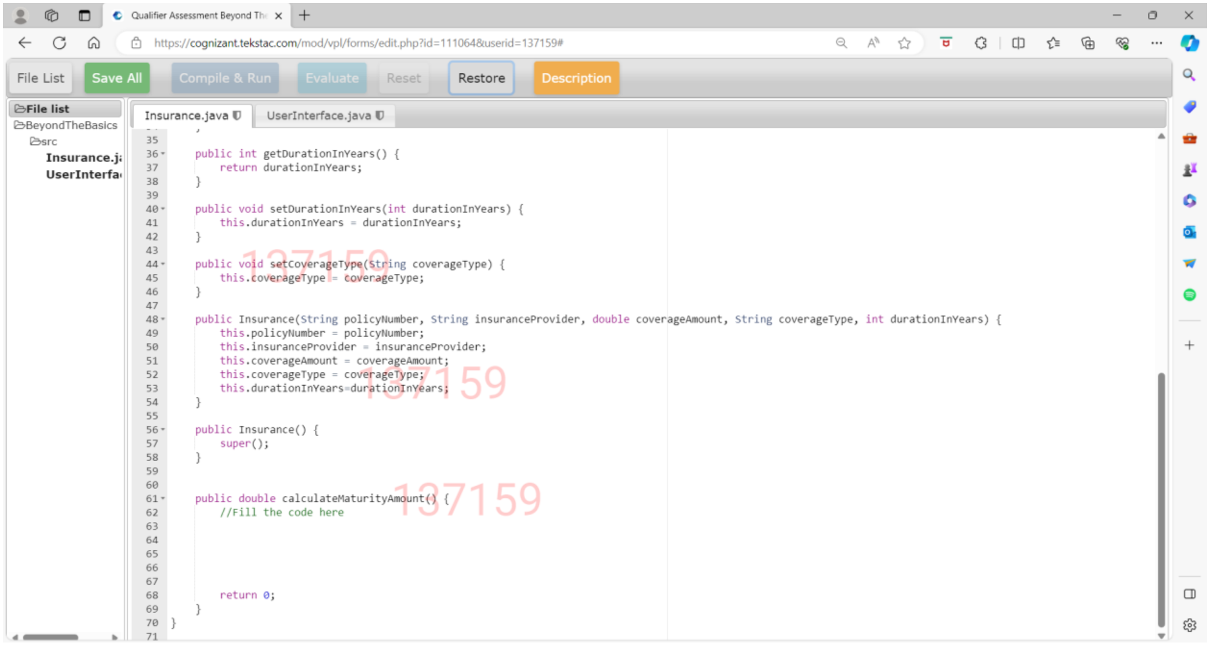
**Sample Output 2**

Invalid details

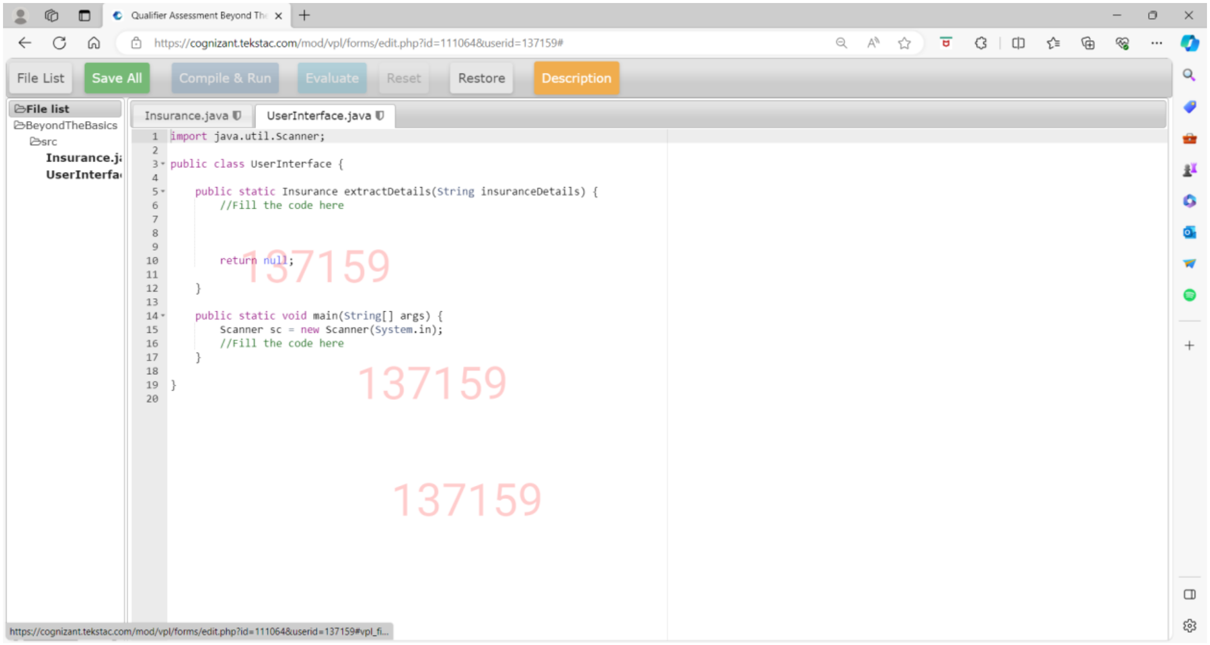
**Explanation :** As coverageAmount is invalid.



```java
public class Insurance {
    private String policyNumber;
    private String insuranceProvider;
    private double coverageAmount;
    private String coverageType;
    private int durationInYears;

    public String getPolicyNumber() {
        return policyNumber;
    }

    public void setPolicyNumber(String policyNumber) {
        this.policyNumber = policyNumber;
    }

    public String getInsuranceProvider() {
        return insuranceProvider;
    }

    public void setInsuranceProvider(String insuranceProvider) {
        this.insuranceProvider = insuranceProvider;
    }

    public double getCoverageAmount() {
        return coverageAmount;
    }

    public void setCoverageAmount(double coverageAmount) {
        this.coverageAmount = coverageAmount;
    }

    public String getCoverageType() {
        return coverageType;
    }

    public int getDurationInYears() {
        return durationInYears;
```



```java

    public int getDurationInYears() {
        return durationInYears;
    }

    public void setDurationInYears(int durationInYears) {
        this.durationInYears = durationInYears;
    }

    public void setCoverageType(String coverageType) {
        this.coverageType = coverageType;
    }

    public Insurance(String policyNumber, String insuranceProvider, double coverageAmount, String coverageType, int durationInYears) {
        this.policyNumber = policyNumber;
        this.insuranceProvider = insuranceProvider;
        this.coverageAmount = coverageAmount;
        this.coverageType = coverageType;
        this.durationInYears=durationInYears;
    }

    public Insurance() {
        super();
    }


    public double calculateMaturityAmount() {
        //Fill the code here





        return 0;
    }
}

```


Qualifier Assessment Beyond The
https://cognizant.tekstac.com/mod/vpl/forms/edit.php?id=111064&userid=137159#
File List
Save All
Compile & Run
Evaluate
Reset
Restore
Description
File list
BeyondTheBasics
src
Insurance.ja
UserInterfa
Insurance.java
UserInterface.java
https://cognizant.tekstac.com/mod/vpl/forms/edit.php?id=111064&userid=137159#vpl_fi...

```
import java.util.Scanner;

public class UserInterface {

    public static Insurance extractDetails(String insuranceDetails) {
        //Fill the code here



        return null;

    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        //Fill the code here
    }

}

```