**Assignment**

1. **Write a program to calculate the Simple Interest with minimal code using features of Java 11.**

**Hint: Use the concept of functional interface and Local variable syntax for lambda parameters**

**Code:**

**package** Abstraction;

**import** java.util.\*;

**interface** SimpleInterest{

**public** **double** simple(**double** principal, **double** time, **double** rate);

}

**public** **class** CollectionQ1 {

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

SimpleInterest simpleInterest = ((principal, time, rate) -> (principal\*time\*rate)/100);

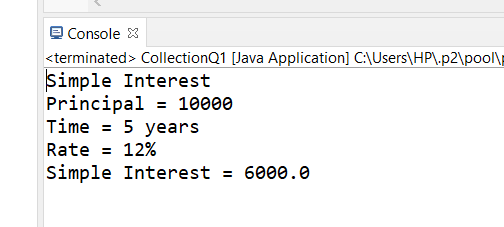
System.***out***.println("Simple Interest\n"+"Principal = 10000\n"+"Time = 5 years\n"+"Rate = 12%");

System.***out***.println("Simple Interest = "+simpleInterest.simple(10000, 5, 12));

}

}

**Output:**



1. **Java 11 supports var keyword for variable declarations. List the scenarios where var keyword cannot be used for such variable declarations. Give reason in support of your answer for each scenario.**

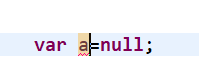
**Answer:**

* You can use var only for local variables (in methods). It cannot be used for instance variables (at class level).
* You cannot use var in Lambda expressions.
* You cannot use var for method signatures (in return types and parameters).

And remember, you cannot use var to declare a variable without explicit initialization, hence the following:



is not allowed, since local variable declaration requires initialization on the right side. That also means this declaration is not valid:



1. **“A quick brown fox jumps over the lazy dog”. Create an ArrayList from the given String. Such an ArrayList should include each word from the given sentence. Finally convert such List to an array using Java 11 methods and print the output.**

**Code:**

package basic;

**import** java.util.ArrayList;

**public** **class** CollectionQ3 {

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

ArrayList<String> arrayList = **new** ArrayList<>();

arrayList.add("A");

arrayList.add("quick");

arrayList.add("brown");

arrayList.add("fox");

arrayList.add("jumps");

arrayList.add("over");

arrayList.add("the");

arrayList.add("lazy");

arrayList.add("dog");

Object[] objects = arrayList.toArray();

**for**(Object obj: objects) {

System.***out***.println(obj+" ");

}

}

}

**Output:**

