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

**interface** Lambda{

**void** SimpleInterest(**int** A,**int** b,**int** C);

}

**public** **class** framework1 {

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

Lambda obj;

obj = (A,B,C)-> System.***out***.println(A\*B\*C);

obj.SimpleInterest(50,30,90);

}

}

2. 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

**class** Demo1 {

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

{

var x = 100;

var y = 1.90;

var z = 'a';

var p = "tanu";

var q = **false**;

System.***out***.println(x);

System.***out***.println(y);

System.***out***.println(z);

System.***out***.println(p);

System.***out***.println(q);

}

}

Output

100

1.9

a

tanu

**false**

2. var can be used in a local variable declaration.

**class** Demo2 {

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

{

var x = 100;

System.***out***.println(x);

}

}

Output

100

3. var cannot be used in an instance and global variable declaration.

**class** Demo3 {

var x = 50;

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

{

System.***out***.println(x);

}

}

Output

prog.java:8: error: 'var' is not allowed here

var x = 50;

^

1 error

4. var cannot be used as a Generic type.

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

**class** Demo4 {

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

{

var<var> al = **new** ArrayList<>();

al.add(10);

al.add(20);

al.add(30);

System.***out***.println(al);

}

}

Output

prog.java:10: error: 'var' is not allowed here

var<var> al = **new** ArrayList<>();

^

1 error

5. var cannot be used with the generic type.

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

**class** Demo5 {

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

{

var<Integer> al = **new** ArrayList<Integer>();

al.add(10);

al.add(20);

al.add(30);

System.***out***.println(al);

var list = **new** ArrayList<String>();

}

}

Output

prog.java:9: error: illegal reference to restricted type 'var'

var<Integer> al = **new** ArrayList<Integer>();

^

1 error

6. var cannot be used without explicit initialization.

**import** java.io.\*;

**class** Demo6 {

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

{

var variable;

var variable = **null**;

}

}

Output

prog.java:13: error: cannot infer type **for** local variable variable

var variable;

^

(cannot use 'var' on variable without initializer)

prog.java:16: error: cannot infer type **for** local variable variable

var variable = **null**;

^

(variable initializer is 'null')

2 errors

7. var cannot be used with Lambda Expression.

Java

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

**interface** myInt {

**int** add(**int** a, **int** b);

}

**class** Demo7 {

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

{

var obj = (a, b) -> (a + b);

System.***out***.println(obj.add(2, 3));

}

}

Output

prog.java:13: error: cannot infer type **for** local variable obj

var obj = (a, b) -> {

^

(lambda expression needs an explicit target-type)

1 error

8. var cannot be used **for** method parameters and **return** type.

**class** Demo8 {

var method1() { **return** ("Inside Method1"); }

**void** method2(var a) { System.out.println(a); }

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

{

Demo1 obj = **new** Demo1();

var res = obj.method1();

obj.method2();

}

}

Output

prog.java:6: error: 'var' is not allowed here

var method1()

^

prog.java:11: error: 'var' is not allowed here

**void** method2(var a)

^

2 errors

3. “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.

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** framework3 {

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

String string ="A quick brown fox jumps over the lazy dog";

String[] splitArray = string.split(" ");

List<String> splitList = **new** ArrayList<>();

**for** (String i : splitArray) {

splitList.add(i);

}

System.***out***.println(splitList);

String[] arraySplit = **new** String[splitList.size()];

splitList.toArray(arraySplit);

**for**(String k : arraySplit)

System.***out***.println(k);

}

}