**SAP Id: 500090925 Course: B.Tech(Hons.) CSE DevOps**

**Roll No: R2142210478 Program & Sem: Sem 3**

**Name: MANSI SINGH Session:**

**INDEX**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No.** | **Experiment No.** | **Title** | **Date of Performance** | **Date of Submission** | **Remarks (By Faculty)** |
| **1** | **1** | **BASIC JAVA PROGRAMMING** | **17/08/22** | **24/08/22** |  |
| **2** | **2** | **BASIC JAVA PROGRAMMING** | **17/08/22** | **24/08/22** |  |
| **3** | **3** | **ARRAYS** | **24/8/22** | **7/9/22** |  |
| **4** | **4** | **INHERITANCE** | **7/9/22** |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Note: 1. Submit your original work otherwise you will lose marks.

2. Submit all your experiments into a single file

3. The submission must be properly compiled and in provided format only

**Experiment No. 1\_**

**Date of performance: 17-08-22**

**Date of Submission:24-08-22**

**SAP Id: 500090925**

**Roll No.: R2142210478**

**Name of the Student: MANSI SINGH**

1. Title: Introduction to java environment

2. Objective:

3. List of lab activities:

• Define JDK,JVM,OOPs

• Do the screen clipping of the system path .

4. Algorithm/Flowchart and Code followed by Output screenshot (2samples for each program):

**DEFINING TERMINOLOGIES:**

JDK- The Java Development Kit (JDK) is a cross-platformed software development environment that offers a collection of tools and libraries necessary for developing Java-based software applications and applets. It is a core package used in Java, along with the JVM (Java Virtual Machine) and the JRE (Java Runtime Environment).

JDK=JRE+Development Tools

JDK contains:

• Java Runtime Environment (JRE),

• An interpreter/loader (Java),

• A compiler (javac),

• An archiver (jar) and many more.

JVM-

• A specification where working of Java Virtual Machine is specified. But imple-mentation provider is independent to choose the algorithm. Its implementation has been provided by Oracle and other companies.

• An implementation Its implementation is known as JRE (Java Runtime Envi-ronment).

• Runtime Instance Whenever you write java command on the command prompt to run the java class, an instance of JVM is created.

The JVM performs following operation:

o Loads code

o Verifies code

o Executes code

o Provides runtime environment

JVM provides definitions for the:

o Memory area

o Class file format

o Register set

o Garbage-collected heap

o Fatal error reporting etc.

JRE- A JRE is made up of a Java virtual machine (JVM), Java class libraries, and the Java class loader. JDKs are used to develop Java software; JREs provide programming tools and de-ployment technologies; and JVMs execute Java programs.

In order for software to execute a program, it needs an environment to run in—usually an oper-ating system (OS) like Linux, Unix, Microsoft Windows, or MacOS. With no other supporting environments, programs are constrained by the capabilities of the OS and it’s resources (such as memory and program files).

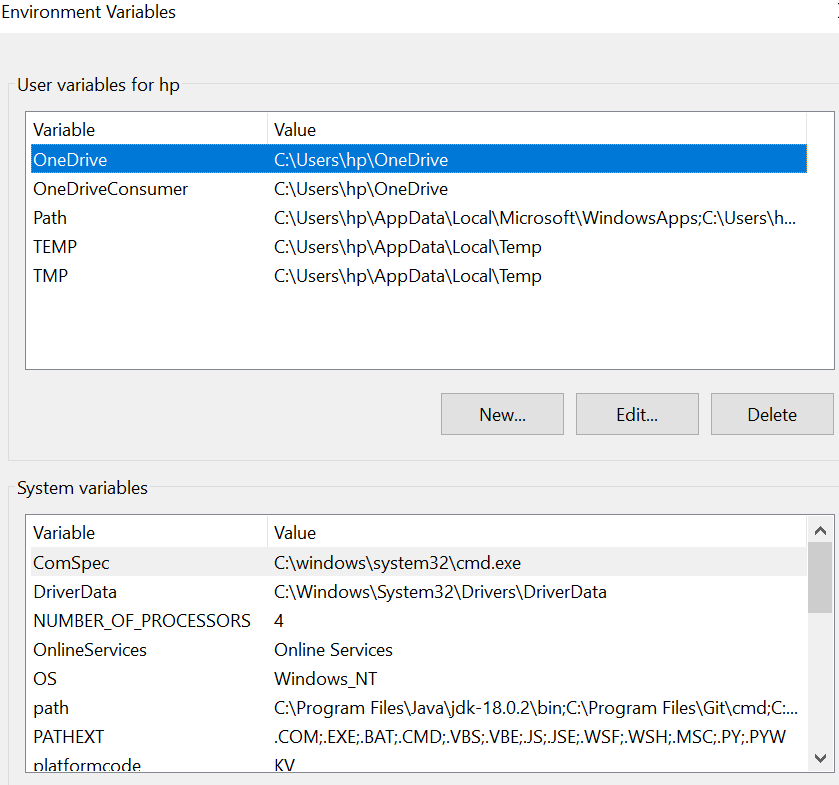
A JRE acts as a kind of translator and facilitator between the Java program and the OS.

Javac- Javac is the standard Java compiler and part of the Java Development Kit. It creates bytecode for the Java virtual machine from valid Java code. While it is primarily run from the command line, it can run programmatically using the Java compiler API.

SDK- A Software Development Kit, or an SDK, is a collection of tools that you need to devel-op an application for a specific software framework. For example, to develop applications in Java, you need a Java SDK (JDK). SDKs contain binaries, source code for the binaries, and documentation for the source code. JDK builds also contain annotations.

Generally, SDKs are global. It means that one SDK can be used in multiple projects and mod-ules. After you create a new project and define an SDK for it, you can configure modules in this project to inherit its SDK. You can also specify an SDK for each module individually. For more information, refer to Change module SDK.

SCREENSHOT OS SYSTEM PATH:



Brief note:

Learned about JVM, JRE, JDK and how to set path.

**Experiment No. \_\_2\_**

**Date of performance:17/08/22**

**Date of Submission:24/08/22**

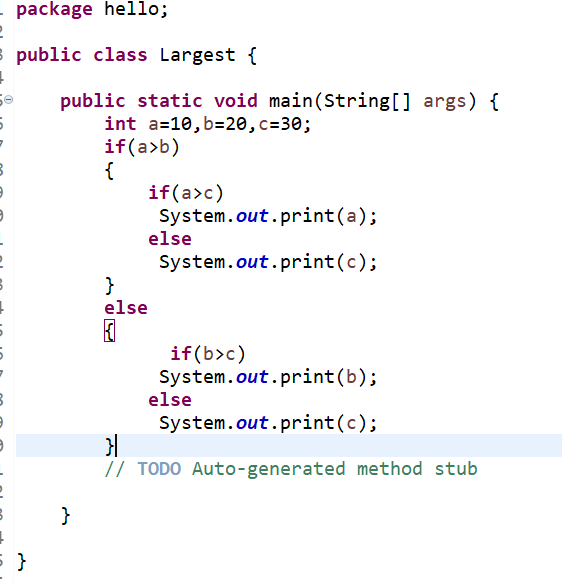
**SAP Id:500090925**

**Roll No.: R2142210478**

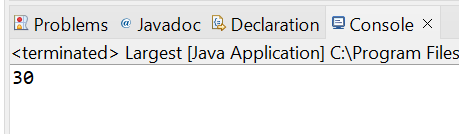
**Name of the Student: Mansi Singh**

1. **Title: Basic JAVA Programming**
2. **Objective: Basic JAVA Programming**
3. **List of lab activities:**
4. Write a program to find largest of 3 numbers.
5. Write a program to implement a command line calculator.
6. Write a program to print grades.
7. **Algorithm/Flowchart and Code followed by Output screenshot (2samples for each program):**
8. **Algorithm:**
9. Start.
10. Take three numbers in **a**, **b**, **c**.
11. Check if **a** is greater than **b**.
12. If above condition is true, go to ***step 5***, else go to ***step 7***.
13. Check if **a** is greater than **c**.
14. If above condition is true, **a** is the largest, else **c** is the largest. Go to ***step 9***.
15. Check if **b** is greater than **c**.
16. If above condition is true, **b** is the largest, else **c** is the largest.
17. Stop.

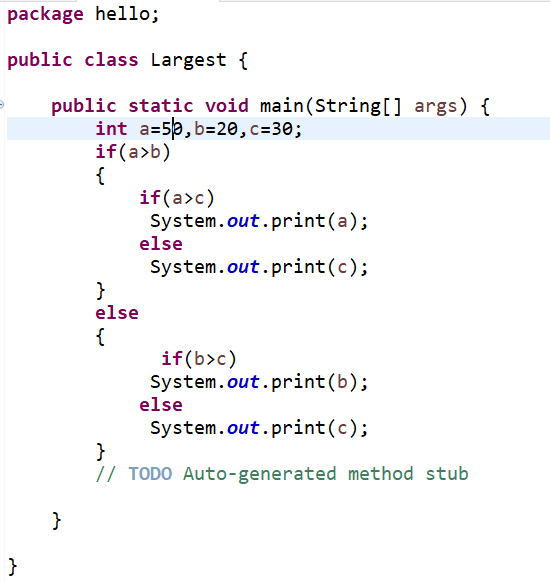
**Code:**

****

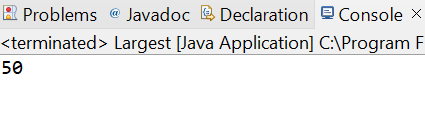
**Output:**

****

**Code:**

****

**Output:**

****

**2] Write a program to implement a command line calculator.**

**Algorithm:**

The args array will hold the values passed on the commad line. args[0] is the first element of the array.

Integer.parseInt can convert a string into an integer number.

int is used to decalare a variable as integer.

String is used to declare a variable as string.

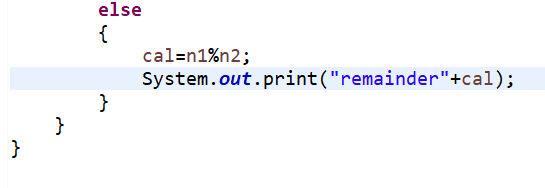
if has its condition in parentheses and the code in curly braces.

else if allows us to attach more, alternative conditions.

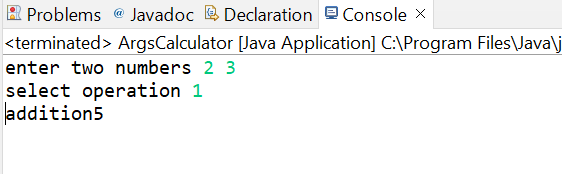
throw new java.lang.Error("...") will raise an exception in case the user supplied an operator we don't handle.

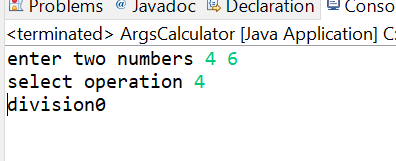
Code:





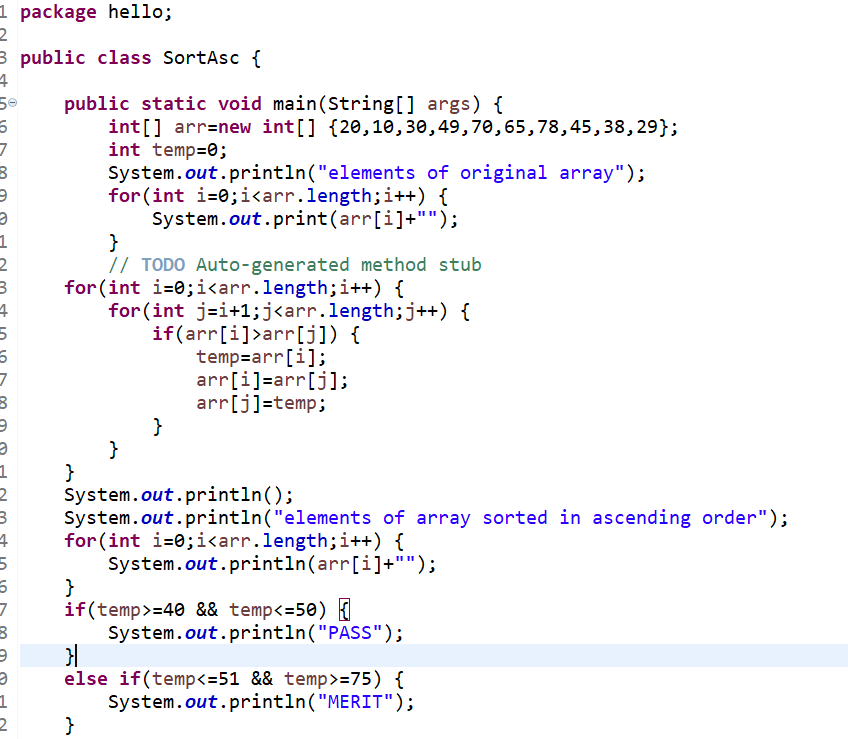
Output:

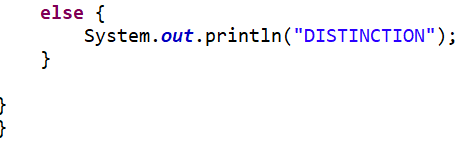




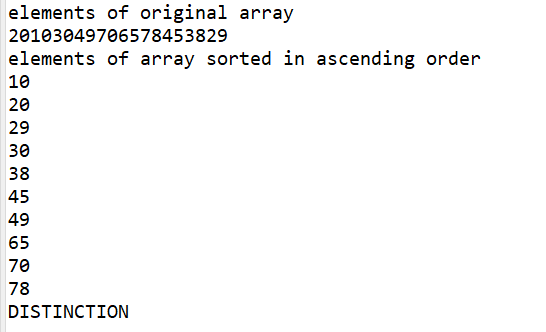
1. Write a program to accept 10 student’s marks in an array, arrange it into ascending order, convert into the following grades and print marks and grades in tabular form.

CODE:





OUTPUT:



Brief note:

Learned basic java programming

**Experiment No. \_\_3\_**

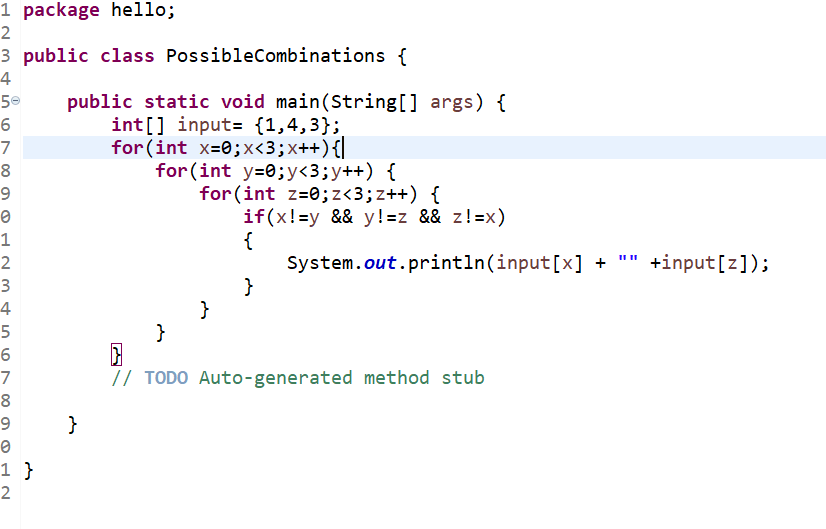
**Tiltle: Arrays**

**1]Write a program to accept three digits and print all its possible combinations.**

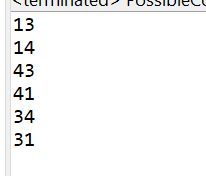
**Algorithm:**

* We have to write 3 for loops and need to ensure all three for loops variable does not have same value.
* Then we can take and print the input array with the three for loop variables to get all different possibles.

**CODE:**

****

**OUTPUT:**

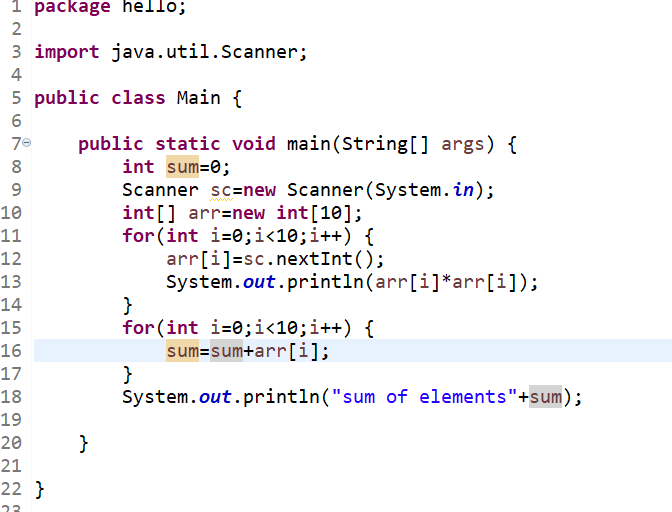
****

**2]Write a java program to accept 10 numbers in an array and compute the square of each number. Print the sum of these numbers**

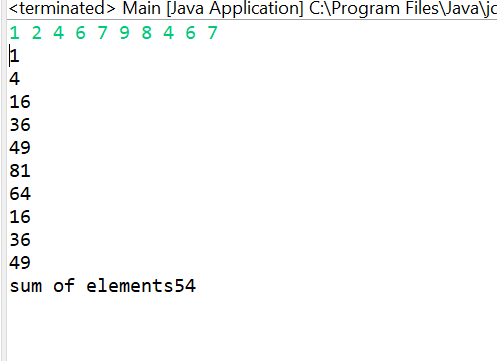
**Algorithm:**

1. Start
2. Declare the array size.
3. Ask the user to initialize the array size.
4. Declare the array.
5. Ask the user to initialize the array elements.
6. Declare a variable sum to store the sum of all the elements in the array.
7. Initialize the variable to 0.
8. Using a for-each loop calculates the sum of all the elements in the array.
9. Display the sum.
10. Stop.

**CODE:**

****

**Output:**

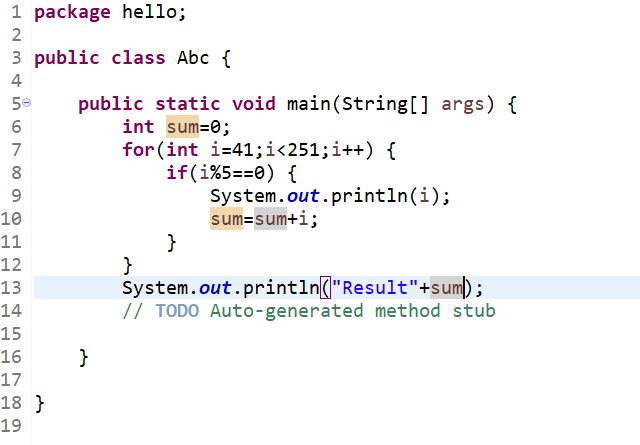
****

**3] Write a program to find the sum of all integers greater than 40 and less than 250 that are divisible by 5**

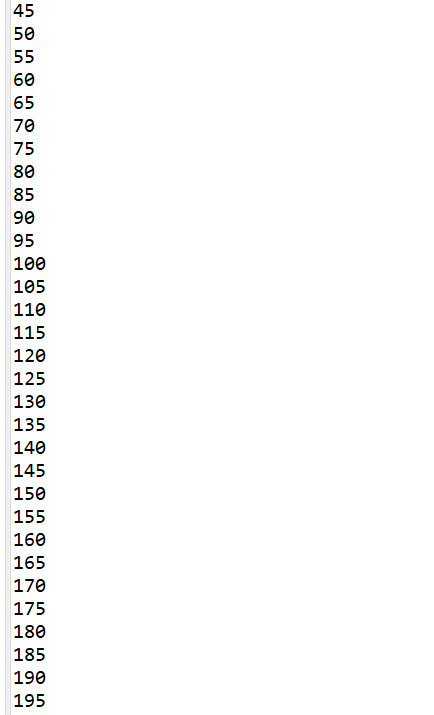
**Algorithm:**

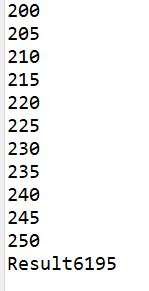
For example, let’s take N = 250 as a limit, then the program should print all numbers less than 250 which are divisible by 5. For this divide each number from 0 to N by 5 and check their remainder. If remainder is 0 in both cases then simply print that number.

**Code:**

****

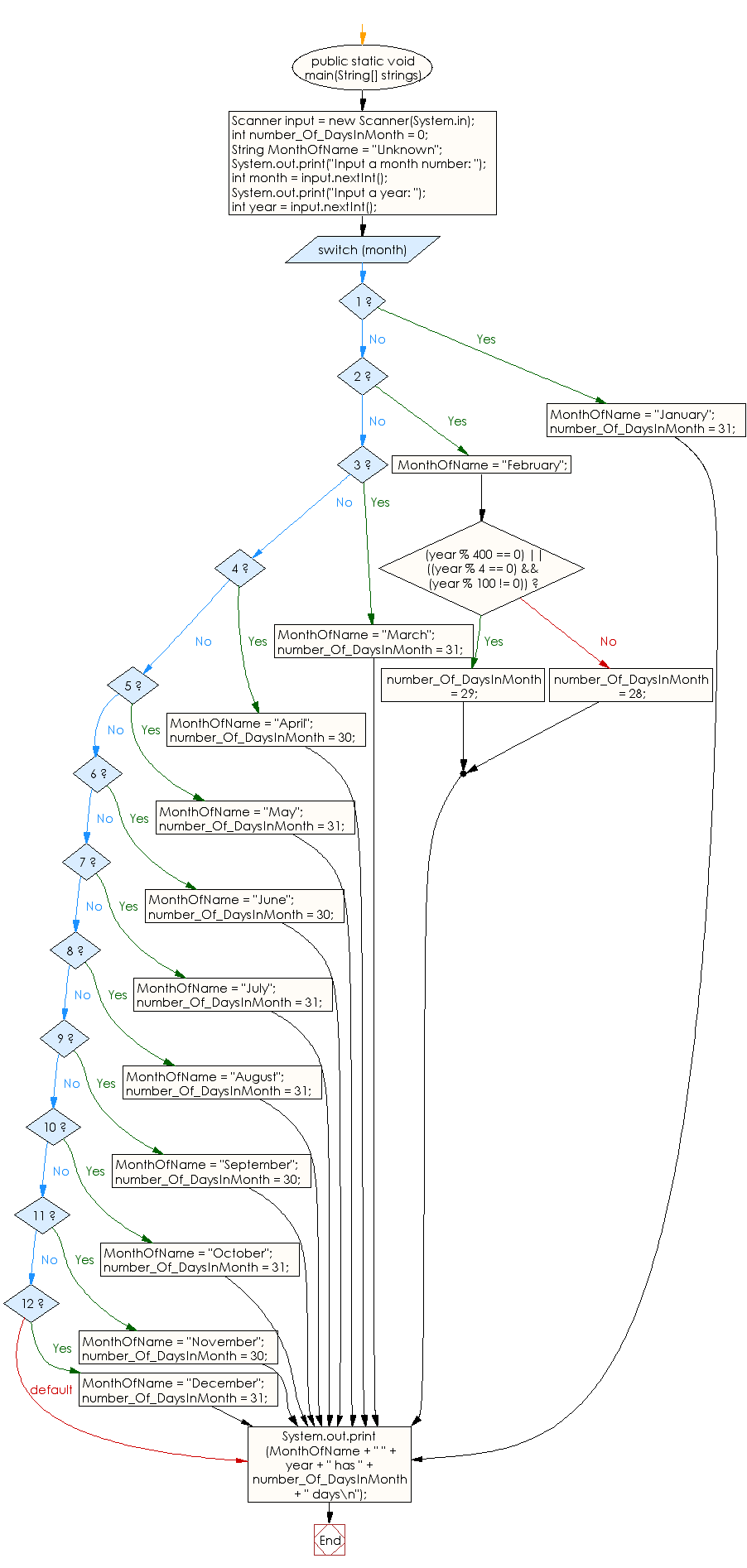
**Output:**

****

****

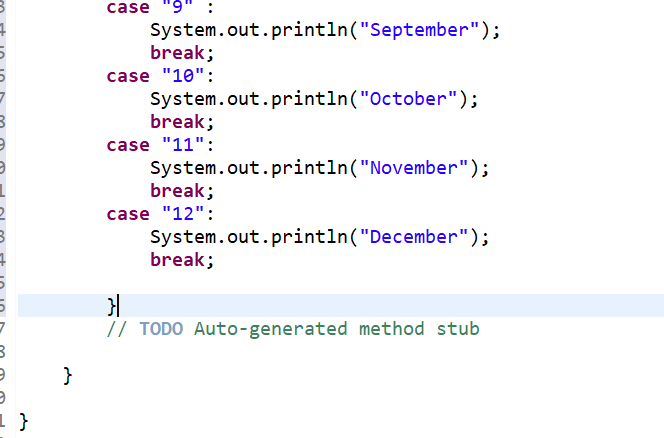
**4] Write a program to input a number of month(1-12) and print its equivalent name of the month.**

**Flowchart:**

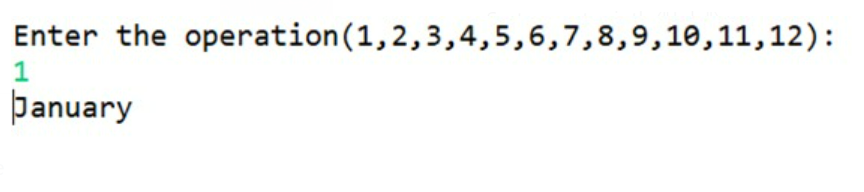


**Code:**

****

****

**Output:**

****

1. **Brief notes about all the concepts related to the lab experiment**

**In experiment number 3, I learned the concept of nested for loop in java, switch case statements in java, Also I learned how to print the square of the elements of the array in java.**

**Experiment No. \_\_4\_**

**TTTLE: Inheritance**

**Objective: Inheritance in Java**

**1] Write a java program to show that private member of a super class cannot be accessed from derived classes.**

**Algorithm:**

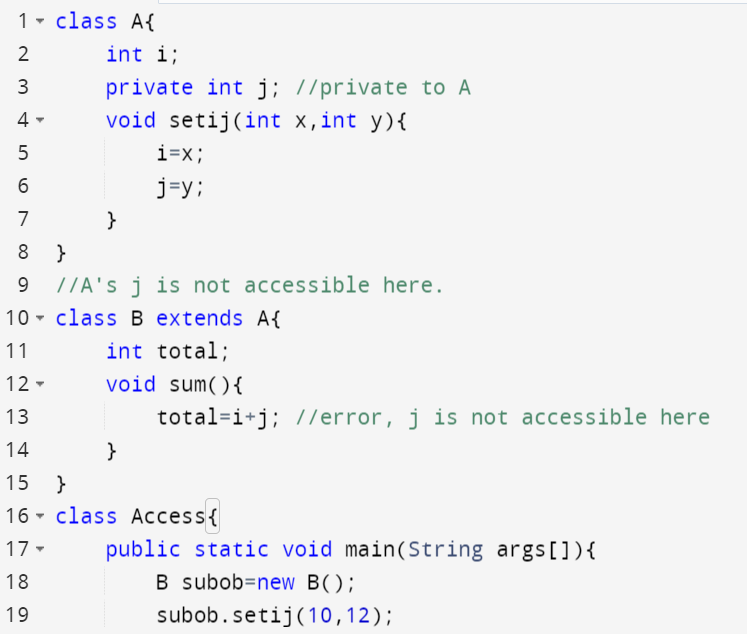
**1] Make Super class**

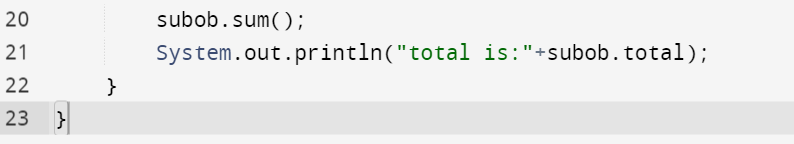
**2] Make private members in this class by using private keyword that is private int l,b;**

**3] Make another class extending Super class**

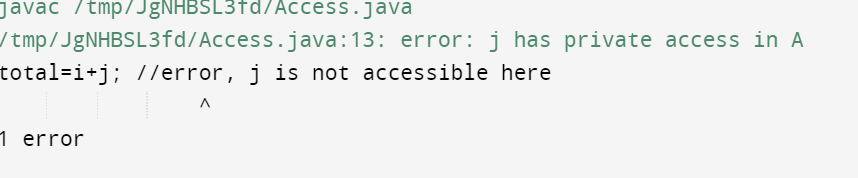
**4] Make object and try to access the private members in Super class.**

**Code:**

****

****

**Output:**

****

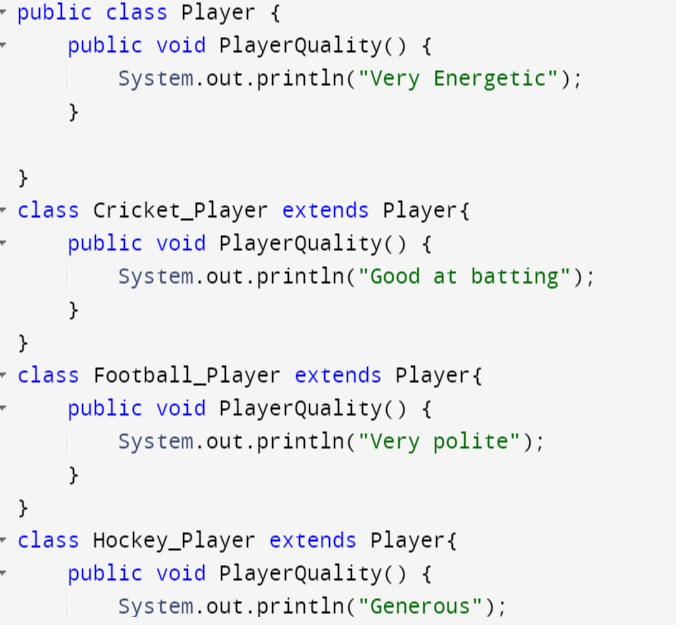
**2]Write a program in java to create a Player class. Inherit the classes Cricket\_Player, Football\_Player and Hockey\_Player from Player class.**

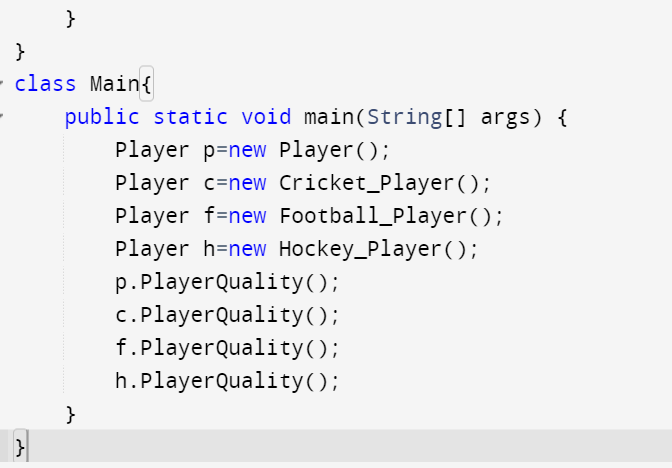
**Algorithm :**

**1] Make class named as Player**

**2] Make another classes with name Cricket\_Player, Football\_Player, and Hockey\_Player inheriting the Player class by using extend keyword.**

**Code:**

****

****

### 3] Write a class Worker and derive classes DailyWorker and SalariedWorker from it. Every worker has a name and a salary rate. Write method ComPay (int hours) to compute the week pay of every worker. A Daily Worker is paid on the basis of the number of days s/he works. The Salaried Worker gets paid the wage for 40 hours a week no matter what the actual hours are. Test this program to calculate the pay of workers. You are expected to use the concept of polymorphism to write this program.

### Algorithm:

### 1] Make class Worker and declare properties of this class such as name empno.

### 2] Make the class DailyWorker and SalariedWorker extending Worker class with method Compay to calculate the salary of the workers.

### CODE:

### 

### 

### OUTPUT:

### 

### 4] Consider the trunk calls of a telephone exchange. A trunk call can be ordinary, urgent or lightning.The charges depend on the duration and type of call. Write a program using the concept of polymorphism in java to calculate the charges.

### Algorithm:

### 1] Make Trunk\_Call class

### 2] Trunk\_Call class will have properties of dur, type and rate

### 3] Make Bill class extending Trunk\_Call class

### 4] Make calculate method for calculating the charges.

### 

### 

### 

### 

### 

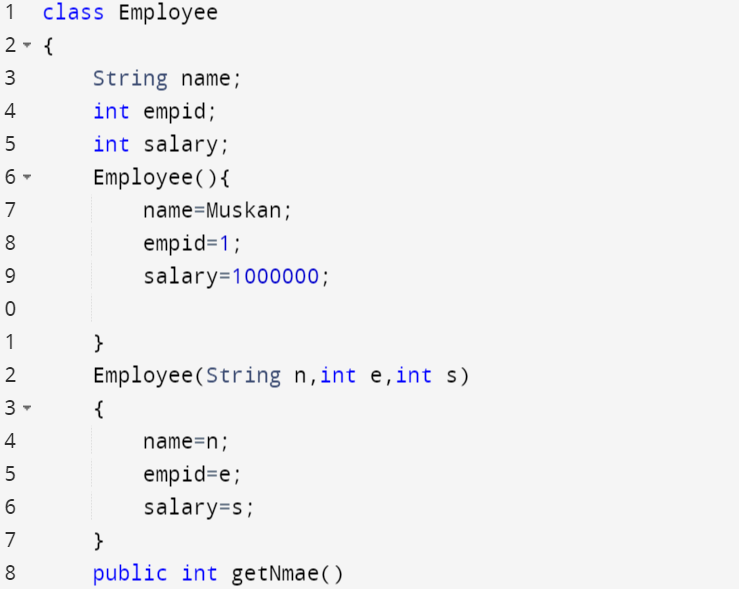
**5] Design class employee of an organization. An employee has a name, empid, and salary. Write the default constructor with parameters(name, empid, and salary) and methods to return name and salary. Also write a method increaseSalary that raises the employee’s salary by a certain user specified percentage. Derive a subclass Manager from employee. Add an instance variable named department to the manager class. Supply a test program that uses these classes and methods.**

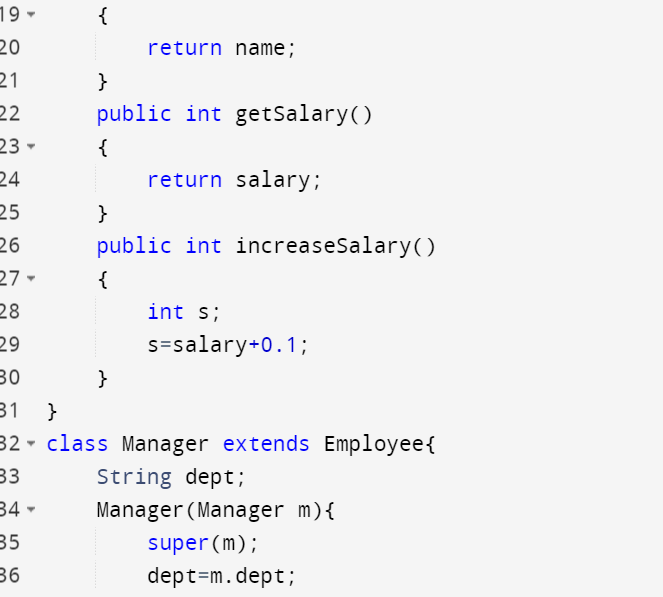
**Algorithm:**

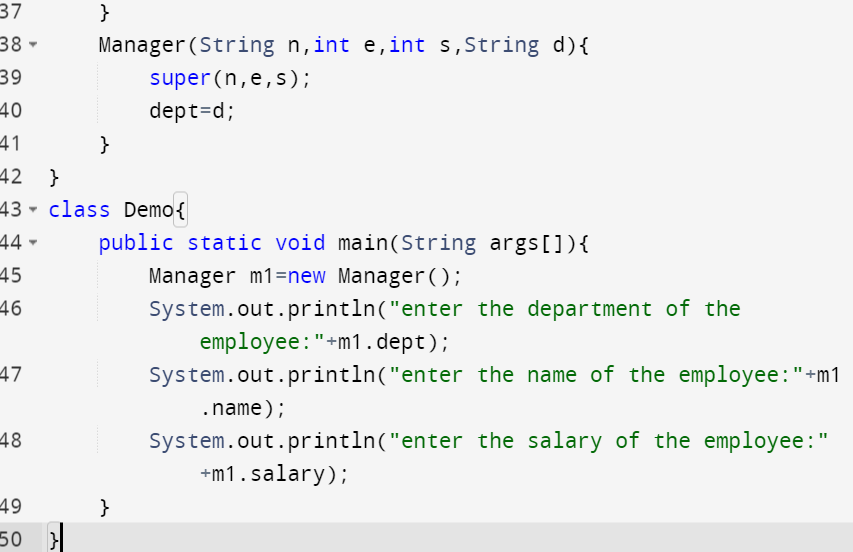
**1]Make class Employee with properties of name empid and salary**

**2] Then make default constructor**

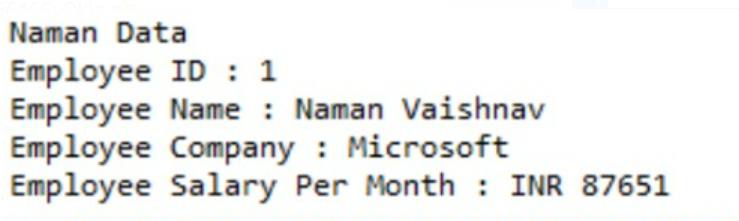
**3] Make method inreaseSalary for calculating increased salary of workers.**

****

****

****

**OUTPUT:**

****

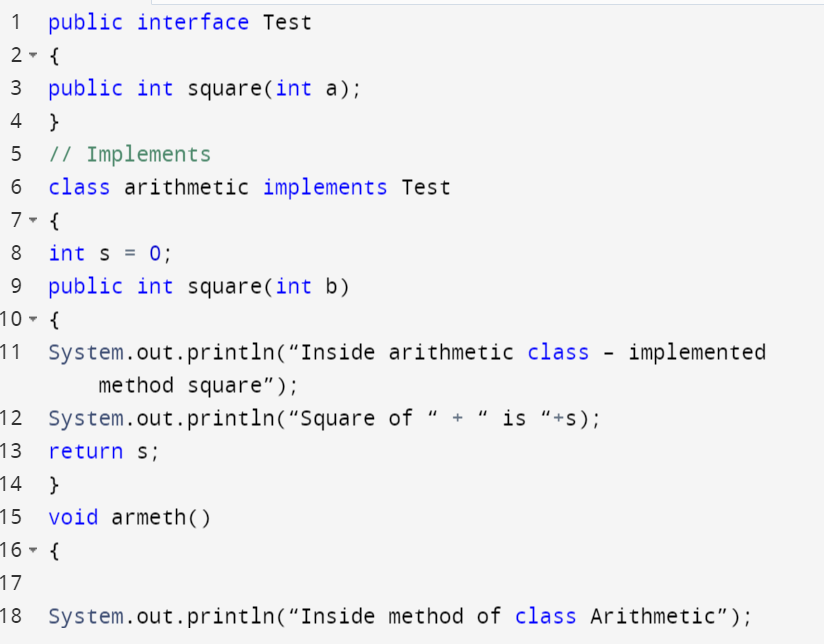
**Brief note:**

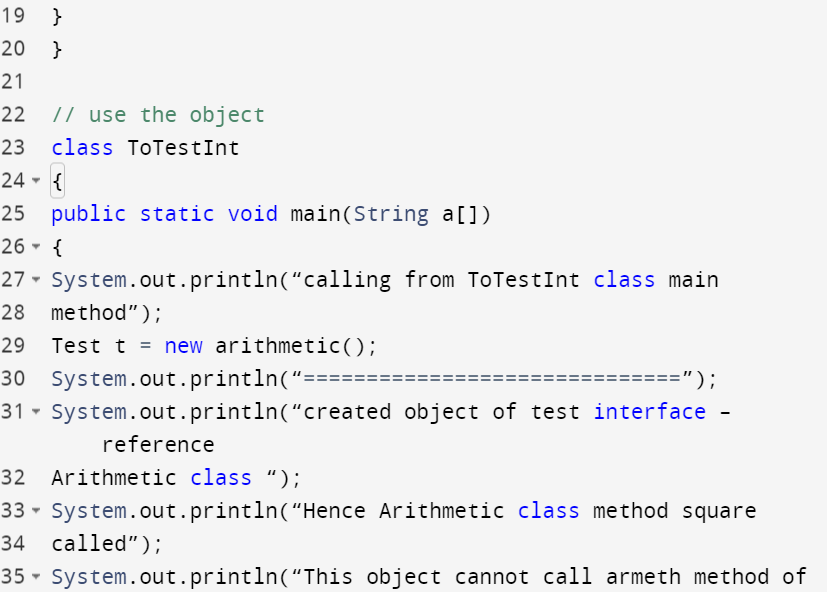
**Learned about inheritance in java**

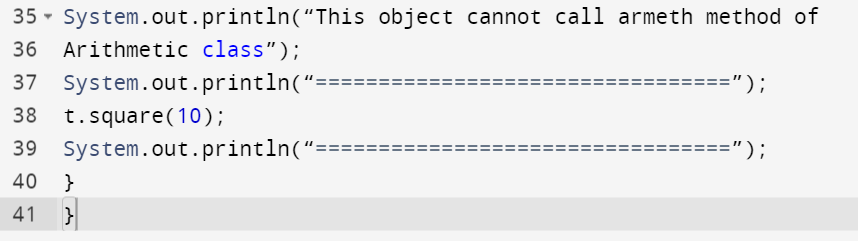
**Experiment 5**

**1] Write a program to create interface named test. In this interface the member function is square. Implement this interface in arithmetic class. Create one new class called to TestInt. In this class use the object of arithmetic class.**

**Code:**

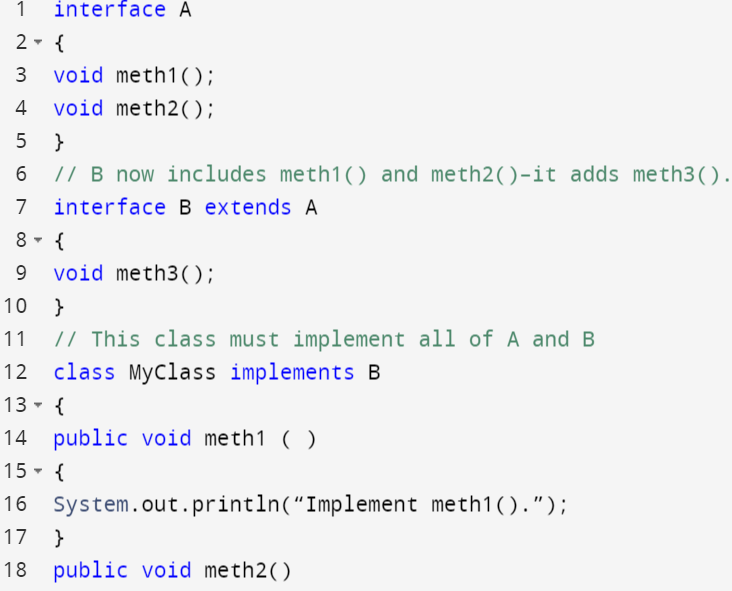
****

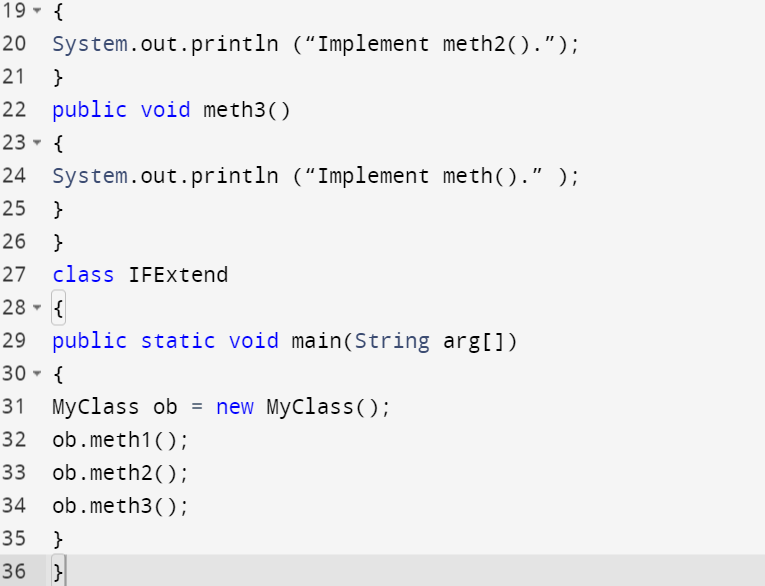
****

****

**2] Write a program to create a interface A, in this interface we have two method meth 1 and meth 2. Implements this interface in another class named Myclass**

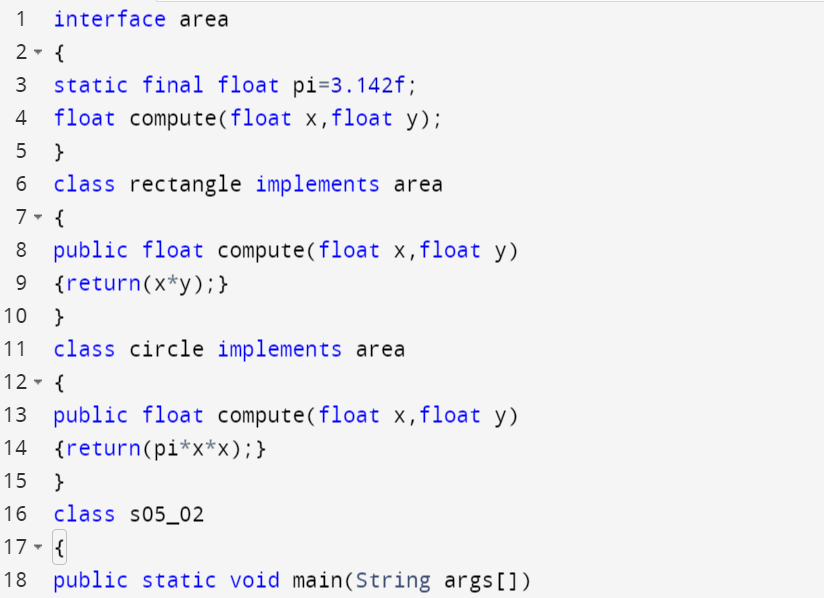
**Code:**

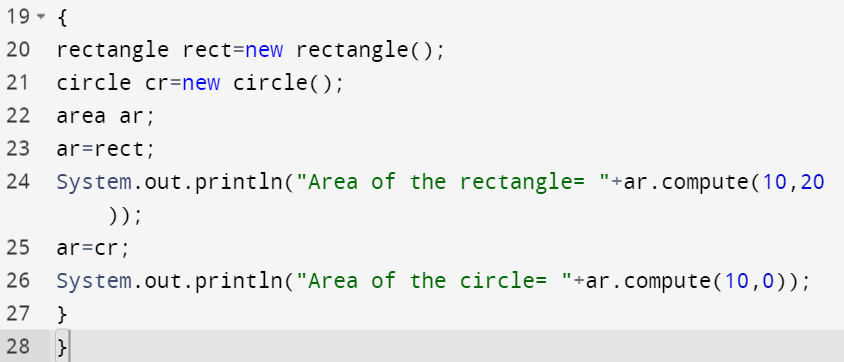
****

****

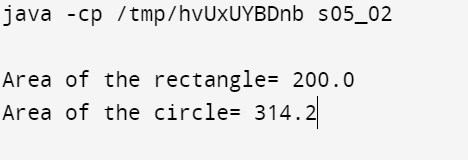
**3] Write a program in java to show the usefulness of interfaces as a place to keep constant value of the program.**

**Code:**

****

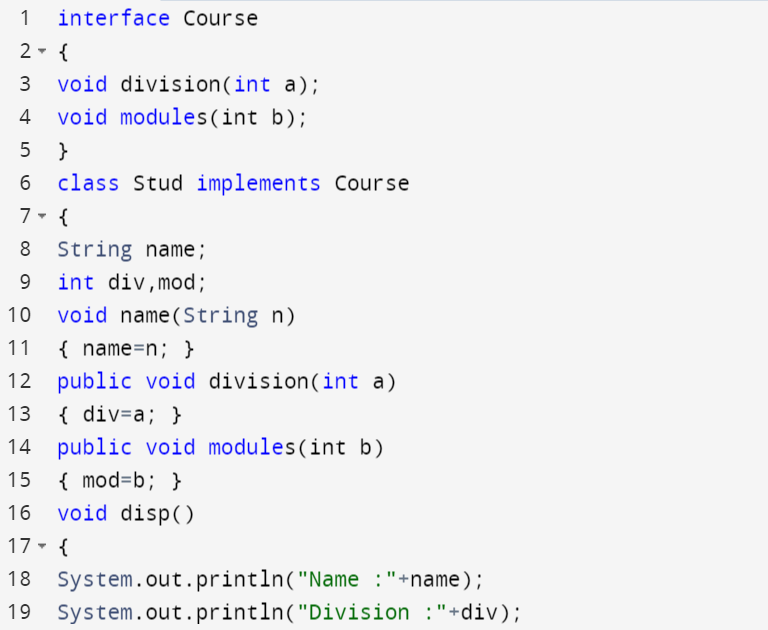
****

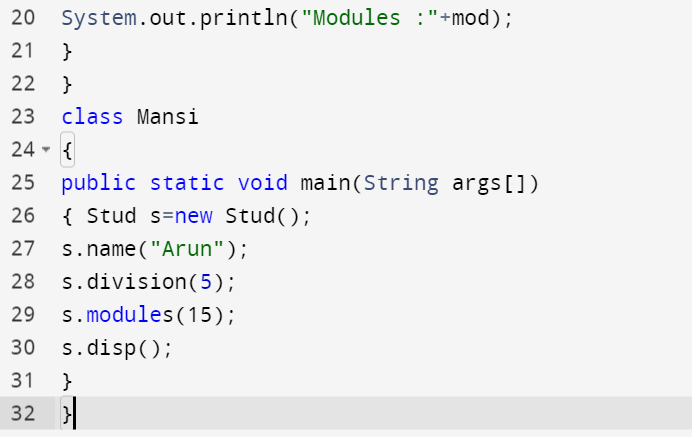
**Output:**

****

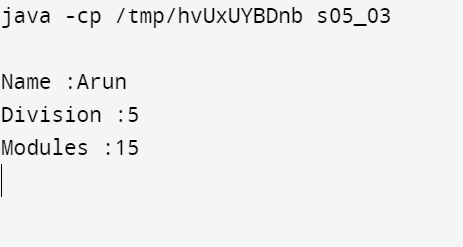
**4] Write a program to create an interface having two methods division and modules. Create a class, which overrides these methods.**

**Code:**

****

****

**Output:**

****

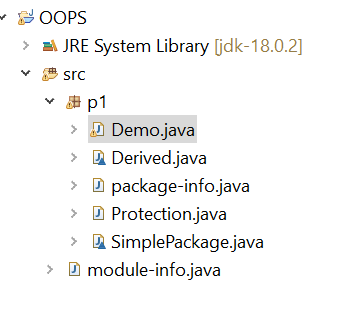
**Experiment 6**

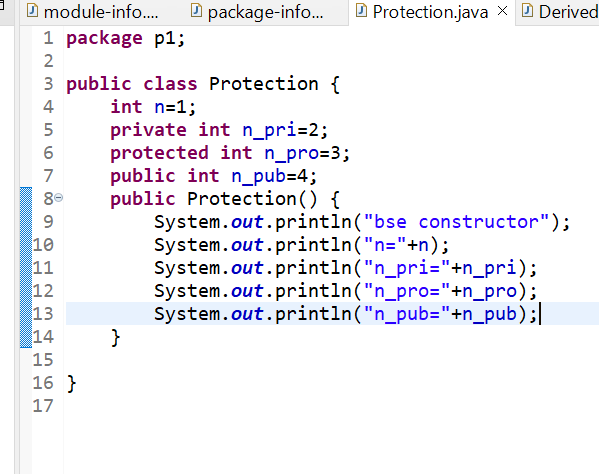
**1] Write a java program to implement the concept of importing classes from user defined package and created packages.**

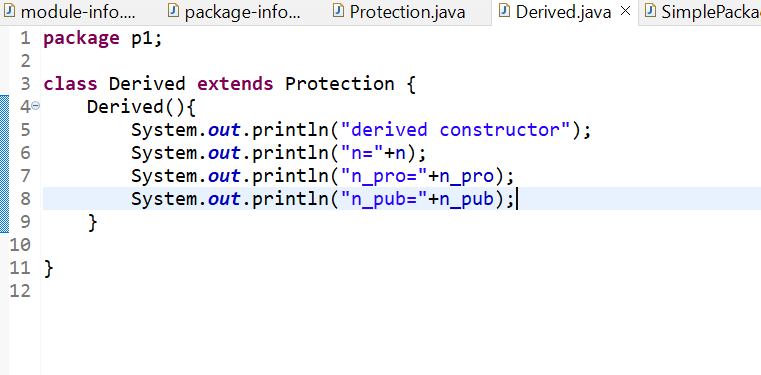
**2] Write a program to make a package Balance. This has an Account class with Display\_Balance method. Import Balance package in another program to access Display\_Balance method of Account class.**

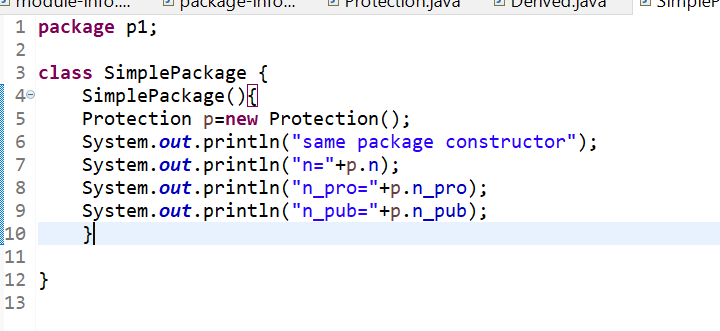
**Solution 1]**

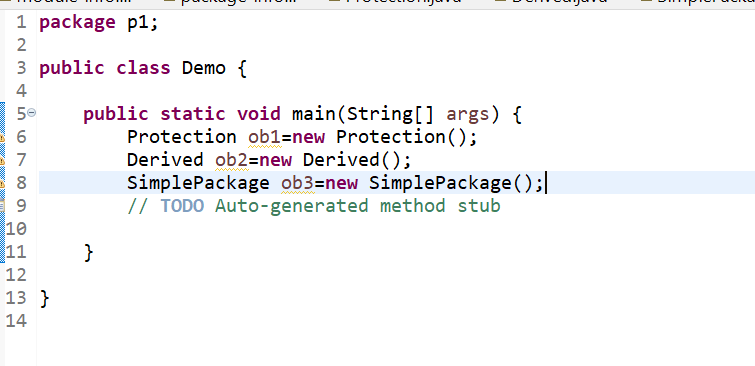
**Code:**

****

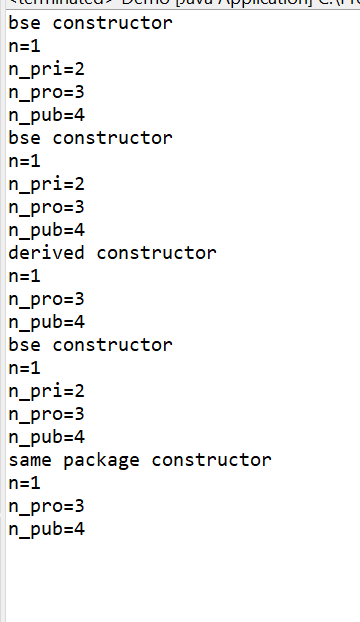
****

****

****

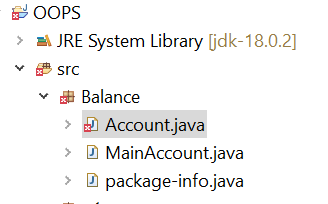
****

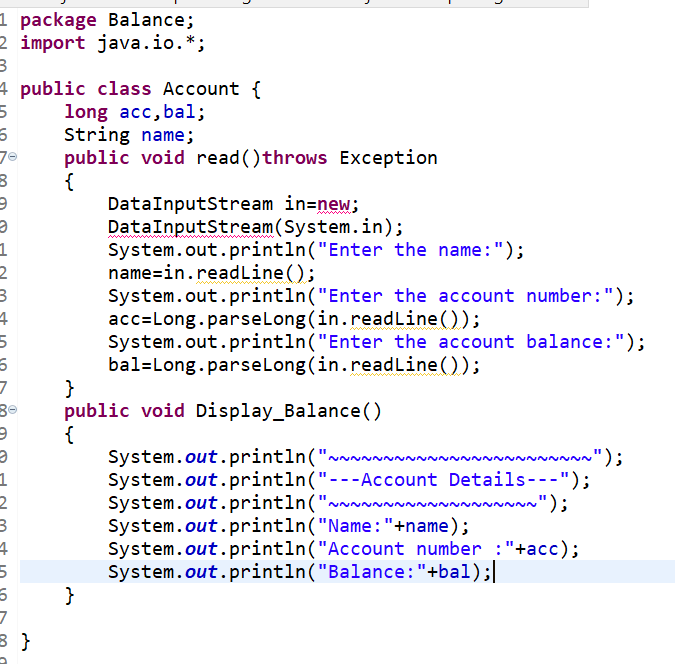
**OUTPUT:**

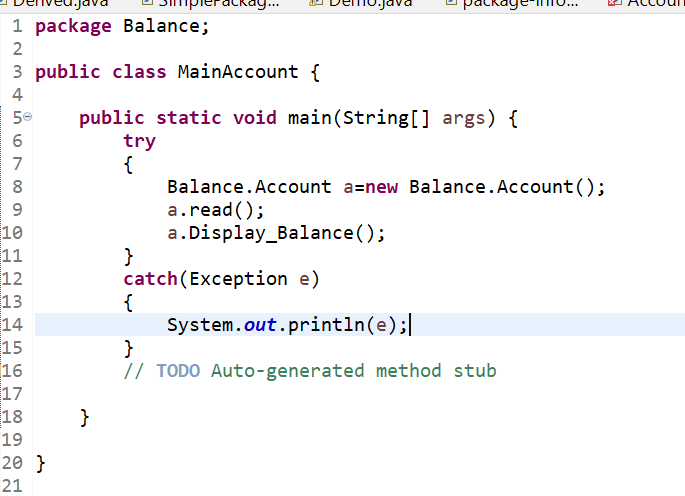
****

**SOUTION2]**

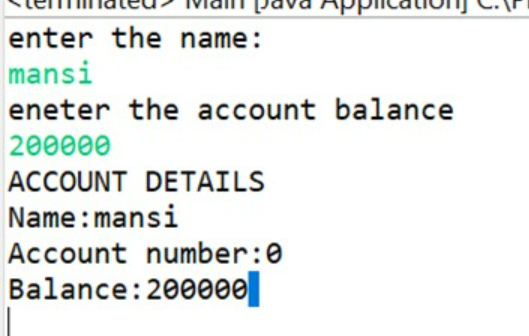
**CODE:**

****

****

****

**OUTPUT:**

****