

LAB-5

Q.1

Create the following interfaces as mentioned:

Reasoning, having a method declaration canReason()

Breathing, having a method declaration canBreath()

Movement, having a method declaration canMove()

Create the following classes:

Human

Animal

Plant

Each of these classes will implement a few or all interfaces from the above-mentioned list.

Now, apply your common sense, which class will implement which Interface/ Interfaces. After implementing the Interface, add a meaningful message while overriding the methods.

For example, CanBreath() method must print "Human can breath." for the Human class. In the main class, create Objects of all the classes and call the methods on these objects to demonstrate the example messages.

Q.2

You are given an interface AdvancedArithmetic which contains a method signature `divisor_sum(int n)`. You need to write a class called MyCalculator which implements the interface. `divisorSum` function just takes an integer as input and return the sum of all its divisors. For example divisors of 6 are 1, 2, 3 and 6, so `divisor_sum` should return 12. The value of `n` will be at most 1000.

Read the partially completed code and complete it. You just need to write the MyCalculator class only. Your class shouldn't be public.

```
import java.util.*;
interface AdvancedArithmetic{
    int divisor_sum(int n);
}

//Write your code here

class Solution{
    public static void main(String []args){
        MyCalculator my_calculator = new MyCalculator();
        System.out.print("I implemented: ");
        ImplementedInterfaceNames(my_calculator);
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        System.out.print(my_calculator.divisor_sum(n) + "\n");
        sc.close();
    }
    /*
     * ImplementedInterfaceNames method takes an object and prints
the name of the interfaces it implemented
     */
    static void ImplementedInterfaceNames(Object o){
        Class[] theInterfaces = o.getClass().getInterfaces();
        for (int i = 0; i < theInterfaces.length; i++){
            String interfaceName = theInterfaces[i].getName();
            System.out.println(interfaceName);
        }
    }
}
```

Q.3

Create the following interfaces as mentioned:
Teaching, having a method declaration canTeach()

Research, having a method declaration canResearch()

Evaluate, having a method declaration canEval()

Create the following classes:

Faculty

Scholar

Student

Each of these classes will implement a few or all interfaces from the above-mentioned list.

A faculty can teach, research and evaluate, a scholar can research and evaluate while a student can research.

Implement relevant interfaces as per the information given above. After implementing the Interface, add a meaningful message while overriding the methods.

For example, CanTeach() method must print "Teacher can teach." for the Teacher class. In the main class, create Objects of all the classes and call the methods on these objects to demonstrate the example messages.