**Core Java Assignments**

* Write a program to print factorial of N ( without using any loop)

public class FactorialofN {

public static int factorial(int n) {

if (n == 0)

return 1;

else

return(n \* factorial(n-1));

}

public static void main(String args[]){

int num,fact=1;

int N=5;//It is the number to calculate factorial

fact = factorial(N);

System.out.println("Factorial of N:"+N+" is: "+fact);

}

}

* There is an animal class which has the common characteristics of all animals. Dog, Horse, Cat are animals(sub-class). Each can shout, but each shout is different. Use polymorphism to create objects of same and using an animal variable, make each of the animals shout.

**public class Polymorphism {**

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

**Animal obj1 = new Dog();**

**Dog obj2 = new Dog();**

**Horse obj3 = new Horse();**

**Cat obj4 = new Cat();**

**// obj1.shout(); //output is bark..**

**obj2.shout(); //output is bark..**

**obj3.shout(); //output is bark..**

**obj4.shout(); //output is bark..**

**}**

**}**

**class Animal{**

**public void shout(){**

**System.out.println("Parent animal's shout");**

**}**

**}**

**class Dog extends Animal{**

**public void shout(){**

**System.out.println("bark..");**

**}**

**}**

**class Horse extends Animal{**

**public void shout(){**

**System.out.println("neigh...");**

**}**

**}**

**class Cat extends Animal{**

**public void shout(){**

**System.out.println("meow..");**

**}**

**}**

* Create an abstract class Instrument which is having the abstract function play. Create three more sub classes from Instrument which is Piano, Flute, Guitar. Override the play method inside all three classes printing a message

“Piano is playing tan tan tan tan ” for Piano class

“Flute is playing toot toot toot toot” for Flute class

“Guitar is playing tin tin tin ” for Guitar class

Create an array of 10 Instruments.

Assign different type of instrument to Instrument reference.

Check for the polymorphic behavior of play method.

**abstract class instrument {**

**abstract public void play();**

**}**

**class piano extends instrument {**

**public void play() {**

**System.out.println("Piano is playing tan tan tan tan");**

**}**

**}**

**class flute extends instrument**

**{ public void play() {**

**System.out.println("Flute is playing toot toot toot toot");**

**}**

**}**

**class guitar extends instrument {**

**public void play() {**

**System.out.println("Guitar is playing tin tin tin tin");**

**}**

**}**

**public class inst {**

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

**instrument ins[]=new instrument[10];**

**for(int i=0;i<10;i++) { if(i==1 || i==5 || i==9) ins[i]=new piano();**

**else if(i==3 || i==4 || i==7) ins[i]=new flute();**

**else ins[i]=new guitar();**

**ins[i].play();**

**if(ins[i] instanceof piano) System.out.println("for Piano class");**

**else if(ins[i] instanceof flute) System.out.println("for Flute class");**

**else System.out.println("for Guitar class");**

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

**}**

**}**

**}**

* Write a program to accept name and age of a person from the command prompt(passed as arguments when you execute the class) and ensure that the age entered is >=18 and < 60. Display proper error messages. The program must exit gracefully after displaying the error message in case the arguments passed are not proper. (Hint : Create a user defined exception class for handling errors.)

**import java.util.Scanner;**

**class AgeException extends Exception {**

**public AgeException(String str) {**

**System.out.println(str);}**

**}**

**public class TestDemo {**

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

**Scanner n = new Scanner(System.in);**

**System.out.print("Enter your first name: ");**

**String first\_name = n.next();**

**Scanner s = new Scanner(System.in);**

**System.out.print("Enter ur age :: ");**

**int age = s.nextInt();**

**try {**

**if(age < 18)**

**throw new AgeException("Hi "+first\_name+" your Age is Not accepted");**

**else if(age >60)**

**throw new AgeException("Hi "+ first\_name + " your Age is Not accepted");**

**else**

**System.out.println("Hi "+first\_name+" your Age is accepted");**

**}**

**catch (AgeException a) {**

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

**}**

**}**

**}**

* Create an ArrayList of Employee( id,name,address,sal) objects and search for particular Employee object based on id number and name.

**import java.util.ArrayList;**

**import java.util.Iterator;**

**import java.util.List;**

**class Employee {**

**private int id;**

**private String name;**

**private String address;**

**private Double salary;**

**public Employee(int id, String name, String address, Double salary) {**

**super();**

**this.id = id;**

**this.name = name;**

**this.address = address;**

**this.salary = salary;**

**}**

**public int getId() {**

**return id;**

**}**

**@Override**

**public String toString() {**

**return "Employee [id=" + id + ", name=" + name + ", address=" + address + ", salary=" + salary + "]";**

**}**

**}**

**public class Assignment5 {**

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

**List<Employee> list = new ArrayList<>();**

**list.add(new Employee(101, "Bob", "123 street, Durban", 20000.0));**

**list.add(new Employee(102, "Alice", "234 street, Gouteng", 30000.0));**

**list.add(new Employee(103, "John", "345 street, Pongola", 25000.0));**

**list.add(new Employee(104, "Stuart", "456 street, PineTown", 40000.0));**

**Iterator<Employee> it = list.iterator();**

**int searchId = 103;**

**while (it.hasNext()) {**

**Employee emp = it.next();**

**if (emp.getId() == searchId)**

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

**}**

**}**

**}**