1. Create a teacher class and derive Professor, Associate Professor, Assistant Professor class from Teacher class. Define appropriate constructor for all the classes. Also define a method to display information of Teacher. Make necessary assumptions as required.

```
import java.util.Scanner;
class Teacher {
    String name;
    String department;
    String course;
    String desgination;
    int teacherId;

    public Teacher(String name, String department, String course, String
desgination, int teacherId) {
        this.name = name;
        this.department = department;
        this.course = course;
        this.desgination = desgination;
        this.teacherId = teacherId;
    }
    public void display(){
        System.out.println("Name : " +this.name);
        System.out.println("Department : " +this.department);
        System.out.println("Course : " +this.course);
        System.out.println("Designation : " +this.desgination);
        System.out.println("Teacher id : " +this.teacherId);

    }
}

class Professor extends Teacher{

    public Professor(String name, String department, String course, int
teacherId){
        super(name, department, course, "Professor", teacherId);
    }
}

class Associate_professor extends Teacher{

    public Associate_professor(String name, String department, String
course, int teacherId){
        super(name, department, course, "Associate_professor", teacherId);
    }
}

class Assistant_professor extends Teacher{

    public Assistant_professor(String name, String department, String
course, int teacherId){
        super(name, department, course, "Assistant_professor", teacherId);
    }
}
```

```
public class Teach {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        String name = null; String course = null; String department =null;
        int tid = 0 , choice;

        while(true){
            System.out.println("Select teacher designation : ");
            System.out.println("1.Professor \n2.Assistant Professor
\n3.Associate Professor \n4.Exit");
            System.out.print("Enter your choice : ");
            choice = input.nextInt();
            if (choice != 4) {
                System.out.print("Enter name : ");
                name = input.next();
                System.out.print("Enter course : ");
                course = input.next();
                System.out.print("Enter department : ");
                department = input.next();
                System.out.print("Enter teacher id : ");
                tid = input.nextInt();
            }
            switch(choice) {
                case 1:
                    Professor prof = new Professor(name, course,
department, tid);
                    prof.display();
                    break;
                case 2:
                    Assistant_professor assistant_professor = new
Assistant_professor(name, course, department, tid);
                    assistant_professor.display();
                    break;
                case 3:
                    Associate_professor associate_professor = new
Associate_professor(name, course, department, tid);
                    associate_professor.display();
                    break;
                case 4:
                    System.exit(0);
                    break;
                default:
                    System.out.println("Invalid choice");
            }
        }
    }
}
```

2. An employee works in a particular department of an organization. Every employee has an employee number, name and draws a particular salary. Every department has a name and a head of department. The head of department is an employee. Every year a new head of department takes over. Also, every year an employee is given an annual salary enhancement. Identify and design the classes for the above description with suitable instance variables and methods. The classes should be such that they implement information hiding. You must give logic in support of your design. Also create two objects of each class.

```
class EmployeeData{
    private int emp_id;
    private double emp_salary;
    private String emp_name, department, designation;

    public EmployeeData(int emp_id, double emp_salary, String emp_name,
                        String department, String designation){
        this.emp_id = emp_id;
        this.emp_salary = emp_salary;
        this.emp_name = emp_name;
        this.department= department;
        this.designation = designation;

    }

    public int getEmp_id() {
        return emp_id;
    }

    public void setEmp_id(int emp_id) { this.emp_id = emp_id; }

    public double getEmp_salary() {
        return emp_salary;
    }

    public void setEmp_salary(double emp_salary) {
        this.emp_salary = emp_salary;
    }

    public String getEmp_name() {
        return emp_name;
    }

    public void setEmp_name(String emp_name) {
        this.emp_name = emp_name;
    }

    public String getDepartment() {
        return department;
    }

    public void setDepartment(String department) {
        this.department = department;
    }

    public String getDesignation() {
        return designation;
    }
```

```
 public void setDesignation(String designation) {
        this.designation = designation;
    }
    public void displayEmpDetails(){
        System.out.println("Emp id : " +this.emp_id);
        System.out.println("Emp name : " +this.emp_name);
        System.out.println("Emp salary : " +this.emp_salary);
        System.out.println("Emp department : " +this.department);
        System.out.println("Emp designation : "  +this.designation +"\n");
    }
}

public class Employee {
    public static void main(String[] args){
        EmployeeData emp1 = new EmployeeData(01, 3000000, "A", "IT",
"HOD");
        EmployeeData emp2 = new EmployeeData(02, 2000000, "B", "IT",
"Dean");
        emp2.setEmp_salary(2500000);
        emp1.setEmp_id(10);

        emp1.displayEmpDetails();
        emp2.displayEmpDetails();
    }
}
```

3. Consider a hierarchy, where a sportsperson can either be an athlete or a hockey player. Every sportsperson has a unique name. An athlete is characterized by the event in which he/she participates; whereas a hockey player is characterised by the number of goals scored by him/her. Perform the following tasks using Java :
(I) Create the class hierarchy with suitable instance variables and methods.
(ii) Create a suitable constructor for each class.
(iii) Create a method named display_all_info with suitable parameters. This method should display all theinformation about the object of a class.
(iv) Write the main method that demonstrates polymorphism (Inheritance)

```
class Sportsperson {
    public String name, title;
    void display_data(){
        System.out.println(title +" name : " +name);
    }
    public Sportsperson(String name, String title){
        this.name = name;
        this.title = title;
    }
}
```

```java
class Athlete extends Sportsperson{
    private int noOfEvents;
    public Athlete(String name, int noOfEvents){
        super(name, "Athlete");
        this.noOfEvents = noOfEvents;
    }
    public void setNoOfEvents(int noOfEvents){
        this.noOfEvents = noOfEvents;
    }
    public int getNoOfEvents(){
        return noOfEvents;
    }
    @Override
    void display_data() {
        super.display_data();
        System.out.println("No of events : " +noOfEvents +"\n");
    }
}

class Hockey extends Sportsperson{
    private int noOfGoals;

    public Hockey(String name, int noOfGoals){
        super(name, "Hockey player");
        this.noOfGoals = noOfGoals;
    }
    public int getNoOfGoals(){
        return noOfGoals;
    }
    public void setNoOfGoals(int noOfGoals){
        this.noOfGoals = noOfGoals;
    }

    @Override
    void display_data(){
        super.display_data();
        System.out.println("No of goals : " +noOfGoals +"\n");
    }
}




public class Sports{
    public static void main(String[] args){

        Sportsperson p1 = new Athlete("A", 5);
        Sportsperson p2 = new Hockey("B", 10);

        ((Hockey)p2).setNoOfGoals(15);

        p1.display_data();
        p2.display_data();
    }
}
```

4. Create an interface vehicle and classes like bicycle, car, bike etc, having common functionalities and put all the common functionalities in the interface. Classes like Bicycle, Bike, car etc implement all these functionalities in their own class in their own way (Interface)

```
interface vehicle{
    void gear(int n);
    void increaseSpeed(int n);
    void applyBreak(int n);
    void display();
}
class Bicycle implements vehicle {
    double speed;
    int gear;

    @Override
    public void gear(int n) {
        gear = n;
    }

    @Override
    public void increaseSpeed(int n) {
        speed += n;
    }

    @Override
    public void applyBreak(int n) {
        speed -= n;
    }

    @Override
    public void display(){
        System.out.println("Current speed : " +speed);
        System.out.println("Gear : " +gear);
        System.out.println("Vehicle type : Bicycle");
        System.out.println("No of wheels : 2\n");
    }
}
class Bike implements vehicle {
    double speed;
    int gear;

    @Override
    public void gear(int n) {
        gear = n;
    }

    @Override
    public void increaseSpeed(int n) {
        speed += n;
    }

    @Override
    public void applyBreak(int n) {
        speed -= n;
    }
```

```
    @Override
    public void display(){
        System.out.println("Current speed : " +speed);
        System.out.println("Gear : " +gear);
        System.out.println("Vehicle type : Bike");
        System.out.println("No of wheels : 2\n");
    }
}
class Car implements vehicle {
    double speed;
    int gear;

    @Override
    public void gear(int n) {
        gear = n;
    }

    @Override
    public void increaseSpeed(int n) {
        speed += n;
    }

    @Override
    public void applyBreak(int n) {
        speed -= n;
    }

    @Override
    public void display(){
        System.out.println("Current speed : " +speed);
        System.out.println("Gear : " +gear);
        System.out.println("Vehicle type : Car");
        System.out.println("No of wheels : 4\n");
    }
}

public class VehicleProg {
    public static void main(String[] args){
        Bicycle cycle = new Bicycle();
        cycle.increaseSpeed(10);
        cycle.gear(2);
        cycle.applyBreak(3);
        cycle.display();

        Car car = new Car();
        car.increaseSpeed(60);
        car.gear(4);
        car.applyBreak(10);
        car.display();

        Bike bike = new Bike();
        bike.increaseSpeed(40);
        bike.gear(3);
        bike.increaseSpeed(10);
        bike.display();
    }
}
```

5. Create a class "Amount In Words" within a user defined package to convert the amount into words. (Consider amount not to be more than 100000). **(Packages)**

```
import java.util.Scanner;
public class NumberToWords{
    private static String numbertoword(int a){
        String word="";
        String unitarray[] = {" "," One"," Two"," Three"," Four"," Five",
                              " Six"," Seven"," Eight"," Nine"," Ten",
                               " Eleven"," Twelve"," Thirteen",
                              " Fourteen"," Fifteen"," Sixteen",
                              " Seventeen"," Eighteen"," Nineteen"};

        String tensarray[]={" "," Ten"," Twenty"," Thirty"," Forty",
                        " Fifty"," Sixty"," Seventy"," Eighty"," Ninety"};

        if((a/1000>0)) {
            word = word + numbertoword(a/1000) + " Thousand";
            a=a%1000;
        }
        if((a/100>0)) {
            word = word + numbertoword(a/100) + " Hundred and";
            a=a%100;
        }if((a/20>0)) {
            word = word + tensarray[a/10];
            a=a%10;
        } if(a>0) {
            if(a<20) {
                word = word + unitarray[a]; //numbers from 0 to 19
                System.out.println(word);
            } else {
                word = word + tensarray[a/10];
                System.out.println(word);
            }
        }
        return word;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int number = 0;
        System.out.print("Please enter a number (max upto 100000) : ");
        number = sc.nextInt();
        if(number==0) {
            System.out.println("Zero");
        } else if (number>100000) {
            System.out.println("Number should be less than 100000");
        } else if (number<0) {
            System.out.println("Please enter a positive number");
        } else if (number == 100000){
            System.out.println("Number in words : One hundred thousand");
        }
        else {
            System.out.println("Number in words : " +
                               numbertoword(number));
       }
    }
}
```

6. Write java program where user will enter loginid and password as input. The password should be 8 digit containing one digit and one special symbol. If user enter valid password satisfying above criteria then show “Login Successful Message”. If user enter invalid Password then create InvalidPasswordException stating Please enter valid password of length 8 containing one digit and one Special Symbol. **(Exception Handling)**

```
import java.util.Scanner;
class InvalidPasswordException extends Exception{
    public InvalidPasswordException(String e){
        super(e);
    }
}

public class ExceptionHandling {
    static boolean validatePassword(String password) {
        int len;
        len = password.length();

        int numberCount = 0;
        int specialCharCount = 0;

        char char1[] = password.toCharArray();

        for(int i = 0; i < len; i++) {
            char c = char1[i];
            if(c >= '0' && c <= '9') { numberCount++; }
            if(c == '@' || c == '!') { specialCharCount++; }
        }

        if(len < 8 || numberCount == 0 || specialCharCount == 0) {
            return false;
        }
        else { return true; }
    }

    public static void main(String[] args) throws InvalidPasswordException
{
        String loginId, password;
        Scanner input = new Scanner(System.in);

        System.out.print("Enter login ID : ");
        loginId = input.next();
        System.out.print("Enter password : ");
        password = input.next();

        if (validatePassword(password)){
            System.out.println("Login Successful!");
        }
        else {
            throw new InvalidPasswordException("Please enter valid password
                                                of length 8 containing one
                                        digit and one special symbol.");
       }

    }
}
```

7. Java Program to Create Account with 1000 Rs Minimum Balance, Deposit Amount, Withdraw Amount and Throws LessBalanceException. It has a Class Called LessBalanceException Which returns the Statement that Says WithDraw Amount(_Rs) is Not Valid. It has a Class Which Creates 2 Accounts, Both Account Deposite Money and One Account Tries to WithDraw more Money Which Generates a LessBalanceException Take Appropriate Action for the Same. **(Exception Handling)**

```
import java.util.*;
class LessBalanceException extends Exception{
    public LessBalanceException(double amount){
        System.out.println("WithDraw amount " +amount +" is not valid.");
    }
}

class Account{
    String name;
    int account_number;
    double amount, balance, min_balance = 1000;

    public Account(int account_number, String name, double balance){
        this.account_number = account_number;
        this.name = name;
        this.balance = balance;
    }

    public void setMin_balance(double amount){
        this.min_balance = amount;
    }

    public double getMin_balance(){
        return min_balance;
    }

    public void deposit(double amount){
        balance += amount;
    }

    public void withdraw(double amount) throws LessBalanceException {
        if ((balance-amount)>min_balance){
            balance -= amount;
        }else {
            throw new LessBalanceException(amount);
        }
    }

    public void display_data(){
        System.out.println("Account no : " +account_number);
        System.out.println("Account holder : " +name);
        System.out.println("Current balance : " +balance);
        System.out.println("Minimum balance required : " +min_balance);
    }
}
```

```
public class Bank {
    static int i=0;
    public static void main(String[] args){
        String name;
        int account_number, choice, j = 0, k;
        double amount, balance, min_balance = 1000;
        Scanner input = new Scanner(System.in);
        Account account[] = new Account[10];

        while(true){
            System.out.println("\nBank transactions: ");
            System.out.println("1.Create account \n2.Withdraw \n3.Deposit
                               \n4.Display \n5.Exit");
            System.out.print("Enter your choice : ");
            choice = input.nextInt();

            switch (choice){
                case 1:
                    System.out.print("\nEnter account number : ");
                    account_number = input.nextInt();
                    System.out.print("Enter account-holder's name : ");
                    name = input.next();
                    System.out.print("Enter balance amount : ");
                    balance = input.nextDouble();
                    if (balance<1000){
                        System.out.println("\nInvalid input");
                        System.out.println("Minimum balance should be "
                                           +min_balance +"\n");
                        break;
                    } else {
                        account[i] = new Account(account_number, name,
                                                 balance);
                        i++;
                        break;
                    }

                case 2:
                    System.out.print("\nEnter your account number : ");
                    account_number = input.nextInt();
                    for (k=0; k<=i; k++){
                        if (account_number == account[k].account_number){
                            j=-1;
                            break;
                        }
                    }
                    if (j==-1) {
                        System.out.print("Enter amount to be withdrawn: ");
                        amount = input.nextDouble();
                        try {
                            account[k].withdraw(amount);
                        }catch (LessBalanceException e){
                        }
                    } else {
                        System.out.println("\nInvalid input!");
                        System.out.println("Please enter valid account
                                           number \n");
                    }
                        j=k=0;
                        break;
```

```
                case 3:
                    System.out.print("\nEnter your account number : ");
                    account_number = input.nextInt();
                    for (k=0; k<=i; k++){
                        if (account_number == account[k].account_number){
                            j=-1;
                            break;
                        }
                    }
                    if (j==-1) {
                        System.out.print("Enter amount to be deposited: ");
                        amount = input.nextDouble();
                        account[k].deposit(amount);
                    } else {
                        System.out.println("\nInvalid input!");
                        System.out.println("Please enter valid account
                                            number \n");
                    }
                        j=k=0;
                        break;
                case 4:
                    System.out.print("\nEnter your account number : ");
                    account_number = input.nextInt();
                    for (k=0; k<=i; k++){
                        if (account_number == account[k].account_number){
                            j=-1;
                            break;
                        }
                    }
                    if (j==-1) {
                        account[k].display_data();
                    } else {
                        System.out.println("\nInvalid input!");
                        System.out.println("Please enter valid account
                                            number \n");
                    }
                    j=k=0;
                    break;
                case 5:
                    System.exit(1);
                    break;
                default:
                    System.out.println("\nInvalid input! \n");
            }
        }
    }
}
```

8. Create two threads such that one thread will print even number and another will print odd number in an ordered fashion. **(MultiThreading)**

```
public class Multithread {
    static int n;
    int counter = 1;
    public void printEvenNumbers(){
        synchronized (this){
            while(counter <= n){
                if (counter %2 == 0) {
                    System.out.print(counter + " ");
                    counter++;
                    notify();
                }else {
                    try {
                        wait();
                    } catch (InterruptedException e) {
                        throw new RuntimeException(e);
                    }
                }
            }
        }

    }

    public void printOddNumbers(){
        synchronized(this){
            while(counter <= n){
                if (counter %2 == 1) {
                    System.out.print(counter + " ");
                    counter++;
                    notify();
                }else {
                    try {
                        wait();
                    } catch (InterruptedException e) {
                        throw new RuntimeException(e);
                    }
                }
            }
        }
    }

    public static void main(String[] args){
        Multithread m = new Multithread();
        n=20;
        Thread t1 = new Thread(new Runnable() {
            @Override
            public void run() { m.printOddNumbers(); }
        });

        Thread t2 = new Thread(new Runnable() {
           @Override
           public void run(){ m.printEvenNumbers(); }
        });

        t1.start();
        t2.start();
    }
}
```