**1.**

package com.Practice4;

public class BankAccount {

int accNo;

String custName;

String accType;

float balance;

public BankAccount(int accNo, String custName, String accType, float balance) {

super();

this.accNo = accNo;

this.custName = custName;

this.accType = accType;

this.balance = balance;

}

public void deposit(float amt) throws NegativeAmount {

try {

if(amt<0 || balance < 0) {

throw new NegativeAmount("negative amount deposit is not acceptable");

}else {

balance=balance+amt;

}

}catch(NegativeAmount ne) {

System.***err***.println(ne.getMessage());

}

}

public float getBalance() throws LowBalanceException{

try {

if(balance<1000 && accType=="saving" || balance <5000 && accType=="current") {

throw new LowBalanceException("The balance is low");

}else {

return balance;

}

}catch(LowBalanceException l) {

System.***err***.println(l.getMessage());

}

return balance;

}

}

package com.Practice4;

public class LowBalanceException extends Exception {

public LowBalanceException(String s) {

super(s);

}

}

package com.Practice4;

public class NegativeAmount extends Exception{

public NegativeAmount(String st) {

super(st);

}

}

package com.Practice4;

public class BankAccountApp {

public static void main(String[] args) throws NegativeAmount, LowBalanceException {

// **TODO** Auto-generated method stub

BankAccount ba=new BankAccount(111,"ck","saving",-900);

ba.deposit(1000);

ba.getBalance();

}

}

**2.**

package com.Practice4;

public class CricketRating {

String playerName;

float critic1;

float critic2;

float critic3;

float avgRating;

String coins;

public CricketRating(String playerName, float avgRating) {

super();

this.playerName = playerName;

this.avgRating = avgRating;

}

public CricketRating(String playerName, float critic1, float critic2, float critic3) {

super();

this.playerName = playerName;

this.critic1 = critic1;

this.critic2 = critic2;

this.critic3 = critic3;

}

public CricketRating(String playerName, float critic1, float critic2) {

super();

this.playerName = playerName;

this.critic1 = critic1;

this.critic2 = critic2;

}

public void claculateAverageRating(float critic1,float critic2) {

avgRating=(critic1+critic2)/2;

}

public void calculateAverageRating(float critic1,float critic2,float critic3) {

avgRating=(critic1+critic2+critic3)/3;

}

public String calculateCoins() throws NotEligibleException{

if(avgRating>=7) {

return "gold";

}else if(avgRating>=5 && avgRating<7) {

return "silver";

}else if(avgRating>=2 && avgRating<5) {

return "copper";

}else {

try {

throw new NotEligibleException("Not Eligible");

}catch(NotEligibleException ne) {

System.***err***.println(ne.getMessage());

}

}

return "";

}

}

package com.Practice4;

public class NotEligibleException extends Exception{

public NotEligibleException(String str) {

super(str);

}

}

package com.Practice4;

public class CricketRatingApp {

public static void main(String[] args) throws NotEligibleException {

// **TODO** Auto-generated method stub

CricketRating cr=new CricketRating("john",9.3f,9.67f,8.7f);

cr.calculateAverageRating(9.3f,9.67f,8.7f);

String coins=cr.calculateCoins();

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

CricketRating cr1=new CricketRating("john",1.8f);

String coin=cr1.calculateCoins();

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

}

}

**3.**

package com.Practice4;

public class Applicant {

String applicantName;

String postApplied;

int applicantAge;

boolean flag = true;

public Applicant(String applicantName, String postApplied, int applicantAge) {

super();

this.applicantName = applicantName;

this.postApplied = postApplied;

this.applicantAge = applicantAge;

}

public Applicant(String postApplied, int applicantAge) {

super();

this.postApplied = postApplied;

this.applicantAge = applicantAge;

}

public void isvalidApplicantName(String name) throws InvalidNameException {

try {

if(name==null) {

flag = false;

throw new InvalidNameException("Invalid Applicant Name");

}

}catch(InvalidNameException ine) {

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

}

}

public void isValidPost(String post) throws InvalidPostException{

try {

if(!(post=="Probationary Officer"|| post=="Special Cadre Officer"|| post=="Assistant")) {

flag = false;

throw new InvalidPostException("Invalid post");

}

}catch(InvalidPostException ip) {

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

}

}

public void isValidAge(int age) throws InvalidAgeException {

try {

if(!(age>18 && age<=30)) {

flag = false;

throw new InvalidAgeException("Invalid Age");

}

}catch(InvalidAgeException ia) {

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

}

}

public boolean isFlag() {

return flag;

}

}

package com.Practice4;

public class Validator {

public static void validate(Applicant applicant) throws InvalidAgeException, InvalidPostException, InvalidNameException {

applicant.isvalidApplicantName(applicant.applicantName);

applicant.isValidPost(applicant.postApplied);

applicant.isValidAge(applicant.applicantAge);

if(applicant.isFlag()) {

System.***out***.println("All details are valid");

}

}

}

package com.Practice4;

public class CatheyBankException extends Exception {

public CatheyBankException(String st) {

super(st);

}

}

package com.mphasis.AssignmentDay4.eh3;

public class InvalidAgeException extends CatheyBankException{

public InvalidAgeException(String st) {

super(st);

// **TODO** Auto-generated constructor stub

}

}

package com.mphasis.AssignmentDay4.eh3;

public class InvalidPostException extends CatheyBankException{

public InvalidPostException(String st) {

super(st);

// **TODO** Auto-generated constructor stub

}

}

package com.Practice4;

public class InvalidNameException extends CatheyBankException {

public InvalidNameException(String st) {

super(st);

// **TODO** Auto-generated constructor stub

}

}

package com.mphasis.AssignmentDay4.eh3;

public class ApplicantApp {

public static void main(String[] args) throws InvalidAgeException, InvalidPostException, InvalidNameException {

// **TODO** Auto-generated method stub

Applicant a=new Applicant("Joseph","Probationary Officer",30);

Validator.*validate*(a);

}

}