JAVA EXCEPTION HANDLING ASSIGNMENT 4

1)

**import** java.util.Scanner;

**public** **class** ArithmeticExcDivision

{

**public** **static** **void** main(String arg[])

{

**try**

{

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

System.***out***.println("start program");

System.***out***.println("enter first number:");

**int** first = input.nextInt();

System.***out***.println("enter second number:");

**int** second = input.nextInt();

**int** result = first/second;

System.***out***.println("the result is:" +result);

}

**catch**(ArithmeticException e)

{

System.***out***.println("the error is:" +e);

e.printStackTrace();

}

**finally**

{

System.***out***.println("program ends");

}

}

}

2)

**import** java.util.Scanner;

**public** **class** ArithmeticExcDivision

{

**public** **static** **void** main(String arg[])

{

**try**

{

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

System.***out***.println("start program");

System.***out***.println("enter first number:");

**int** first = input.nextInt();

System.***out***.println("enter second number:");

**int** second = input.nextInt();

**int** result = first/second;

System.***out***.println("the result is:" +result);

}

**catch**(Exception e)

{

**throw** **new** UnsupportedOperationException("division by zero is not possible");

}

**finally**

{

System.***out***.println("program ends");

}

}

}

3)a)

**public** **class** InsufficientBalanceException **extends** Exception

{

**private** **double** amount;

**public** InsufficientBalanceException(**double** amount)

{

**this**.amount = amount;

}

**public** **double** getAmount()

{

**return** amount;

}

}

**class** Account

{

**long** id;

**private** **double** balance;

**public** Account(**int** id)

{

**this**.id = id;

}

**public** **void** deposit(**double** amount)

{

balance += amount;

}

**public** **void** withdraw(**double** amount) **throws** InsufficientBalanceException

{

**if**(amount<balance)

{

balance -= amount;

}

**else**

{

**double** needs = amount-balance;

**throw** **new** InsufficientBalanceException(needs);

}

}

**public** **double** getbalance()

{

**return** balance;

}

}

**public** **class** Main

{

**public** **static** **void** main(String arg[])

{

**int** id;

Account a= **new** Account( id= 1234);

System.***out***.println("account id is:"+id);

a.deposit(2000);

System.***out***.println("depositing rs.2000...");

**try**

{

a.withdraw(2100);

System.***out***.println("withdrawing rs.2100...");

}

**catch**(InsufficientBalanceException ex)

{

System.***out***.println("insufficient balance needs more" +ex.getAmount()+ "to get money");

}

}

}

b)

**import** java.util.Scanner;

**public** **class** IllegalBankTransactionException **extends** Exception

{

**static** **void** amount() **throws** IllegalBankTransactionException

{

**double** balance = 0, deposit, withdraw;

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

System.***out***.println("deposit an amt:");

deposit = input.nextDouble();

balance += deposit;

System.***out***.println("withdraw an amt:");

withdraw = input.nextDouble();

balance -= withdraw;

**if**(withdraw<0)

{

**throw** **new** IllegalBankTransactionException();

}

**else**

System.***out***.println("balance is:" +balance);

}

**public** **static** **void** main(String arg[])

{

**try**

{

*amount*();

}

**catch**(IllegalBankTransactionException ex)

{

System.***out***.println("entering negative amt for withdrawal is not allowed");

ex.printStackTrace();

}

}

}