1. Write an application that accepts tow numbers, divides the first number with the second number throws Arithmetic Exception when divides by zero.

Ans🡪

**import** java.util.Scanner;

**public** **class** ArithmeticException {

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

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

System.***out***.println("Enter Dividend number:");

**int** dividend = sc.nextInt();

System.***out***.println("Enter Divisor number:");

**int** divisor = sc.nextInt();

**int** result = dividend/divisor;

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

sc.close();

}

}

Output🡪

Enter Dividend number:

35

Enter Divisor number:

0

Exception in thread "main" java.lang.ArithmeticException: / by zero

at ArithmeticException.main(ArithmeticException.java:12)

1. Carrying forward with the above problem, handled ArithmeticException by raising UnsupprotedOperationException as a solution.

Ans🡪

**import** java.util.Scanner;

**public** **class** UnsupportedExc {

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

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

**try** {

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

**int** n1= sc.nextInt();

System.***out***.println("Enter the Second number: ");

**int** n2= sc.nextInt();

**int** n3= n1/n2;

System.***out***.println("The ans is:" +n3);

}**catch**( ArithmeticException e) {

e.printStackTrace();7ik

System.***out***.println("exception handled" +e);

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

}

}

}

Output🡪

Enter the first number:

45

Enter the Second number:

0

java.lang.ArithmeticException: / by zero

exception handledjava.lang.ArithmeticException: / by zero

UnsupportedOperationException

at UnsupportedExc.main(UnsupportedExc.java:14)

1. Write an application to perform withraw functionality on savingAccount object. Point to note:
2. Raise InsufficientBalanceException if you are trying to withdraw more than balance or when you balance is zero. E.g if you balance is 2000 and if are trying to withdraw 2100 or if you balance is 0 and you are trying to withdraw positive value.

Ans🡪**package** org.exception.saving;

**import** java.util.Scanner;

**class** InsufficientBalanceException **extends** RuntimeException{

}

**public** **class** SavingAccount {

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

**public** **void** withdrawl(**double** a) {

System.***out***.println("Enter your Id: ");

**long** id =sc.nextLong();

System.***out***.println("Enter your balance:");

**double** b = sc.nextDouble();

**try** {

**if**(a<=b)

{

b = b-a;

System.***out***.println("Balance= " +b);

}

**else**

{

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

}

}

**catch** (InsufficientBalanceException e) {

e.printStackTrace();

}

}

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

{

SavingAccount s = **new** SavingAccount();

s.withdrawl(2000);

}

}

Output:

Enter your Id:

5432

Enter your balance:

200

org.exception.saving.InsufficientBalanceException

at org.exception.saving.SavingAccount.withdrawl(SavingAccount.java:23)

at org.exception.saving.SavingAccount.main(SavingAccount.java:33)

1. Raise IllegalBankTransactionException if you are trying to withdraw a negative value from you balance . E.g. if you try to withdraw a negative value savingACC withdraw(-1000);

Note: SavingAccount

|- Liong id

|- double Balance

|- double withdraw(double amount)

|- double deposite( double amount)

Ans🡪

**package** org.exception.saving;

**import** java.util.Scanner;

**class** IllegalBankTransactionException **extends** RuntimeException{

}

**public** **class** AccountSaving {

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

**public** **void** withdrawl(**double** a) {

System.***out***.println("Enter your Id: ");

**long** id =sc.nextLong();

System.***out***.println("Enter your balance:");

**double** b = sc.nextDouble();

**try** {

**if**(a>0)

{

System.***out***.println("Balance= " +b);

}

**else**

{

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

}

}

**catch** (IllegalBankTransactionException e) {

e.printStackTrace();

}

}

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

{

AccountSaving a = **new** AccountSaving();

a.withdrawl(-10201);

}

}

Output:

Enter your Id:

4523

Enter your balance:

100

org.exception.saving.IllegalBankTransactionException

at org.exception.saving.AccountSaving.withdrawl(AccountSaving.java:23)

at org.exception.saving.AccountSaving.main(AccountSaving.java:33)