Week 10 extra programs :

/\*Implement Interfaces – QUEUE OPERATIONS\*/

import java.util.\*;

interface IntQueue

{

void insert\_rear(int item);

int delete\_front();

void display();

}

class Queue implements IntQueue

{

private int q[];

private int rear;

private int front;

Queue(int size)

{

q = new int[size];

rear = -1;

front = 0;

}

public void insert\_rear(int item)

{

if(rear==q.length-1)

System.out.println("Queue Overflow ");

else

q[++rear] = item;

}

public int delete\_front()

{

if(front>rear)

{

System.out.println("Queue Underflow.");

front = 0;

rear = -1;

return -1;

}

return q[front++];

}

public void display()

{

System.out.println("contents of queue :");

for(int i=front;i<=rear;i++)

System.out.print(q[i]+" ");

System.out.println();

}

}

class Main

{

public static void main(String args[])

{

Queue obj = new Queue(10);

int n,item;

Scanner sc=new Scanner(System.in);

while(true)

{

System.out.println("1.insert into queue \n 2.delete from queue \n 3.display \n Enter 0 to quit ");

n=sc.nextInt();

if(n==0)

break;

switch(n)

{

case 1:System.out.println("enter item ");

item=sc.nextInt();

obj.insert\_rear(item);

break;

case 2:item=obj.delete\_front();

if(item==-1)

System.out.println("queue is empty");

else

System.out.println("deleted item : "+item);

break;

case 3:obj.display();

break;

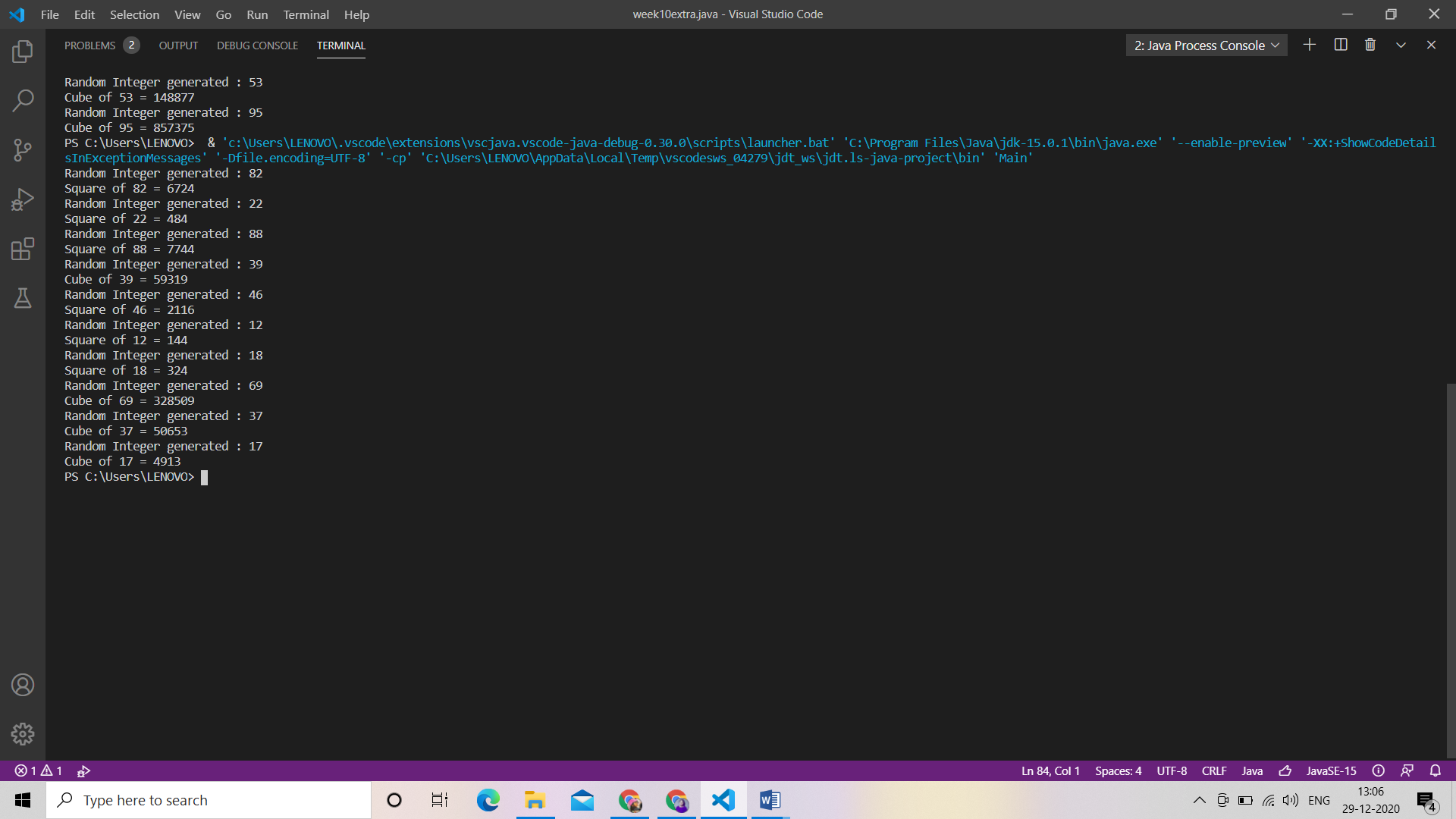
default:System.out.println("invalid choice");

}

}

}

}



/\*2. Write a Java program to compute the factorial of a number. The input value must be tested

for validity. If it is greater than 15, the method ComputeFactorial( ) should raise an

Userdefined Exception MyException with appropriate messages.\*/

import java.util.\*;

class MyException extends Exception

{

int num;

MyException(int a)

{

num=a;

}

public String toString()

{

return("number entered in invalid");

}

}

class factorial

{

static int ComputeFactorial(int n)throws MyException

{

int f=1;

if(n>15)

throw new MyException(n);

else if(n==0)

return 1;

else

{

for(int i=1;i<=n;i++)

f=f\*i;

return(f);

}

}

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.println("enter a number");

int n=sc.nextInt();

int fact=1;

try

{

fact=ComputeFactorial(n);

System.out.println("factorial = "+fact);

}

catch(MyException e)

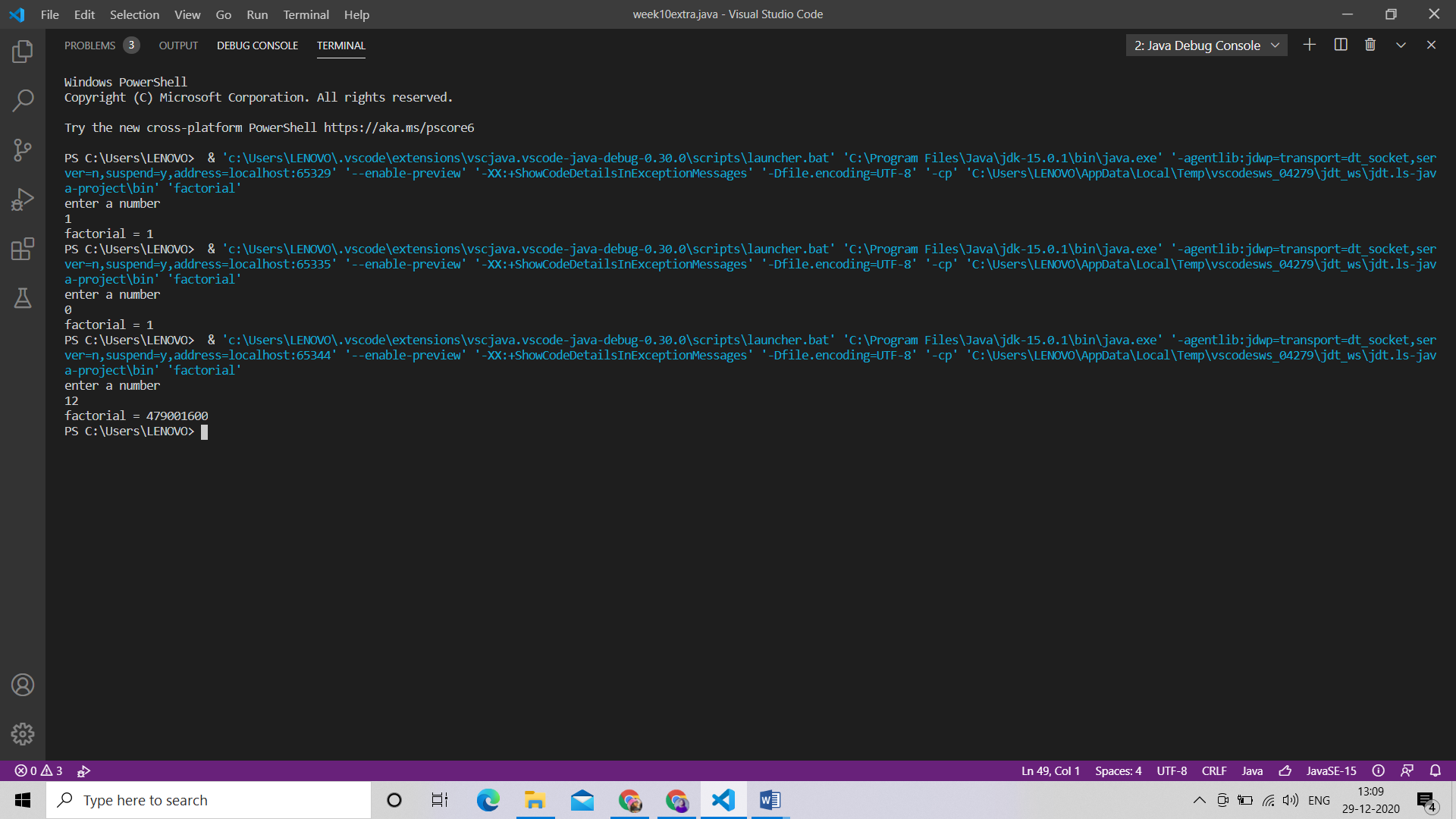
{

System.out.println(e);

}

}

}



/\*3. Write a Java program to create an account class. Define appropriate constructor for this

class. Implement a separate methods to display account balance and withdraw money.

Raise a user defined exception if there is an attempt to withdraw money which is greater

than the account balance. Make necessary assumptions required.\*/

import java.util.\*;

class Insufficient extends Exception

{

double amount;

Insufficient(double a)

{

amount = a;

}

public String toString()

{

return ("INSSUFFICIENT BALANCE IN ACCOUNT /n YOUR ACCOUNT BALANCE="+amount);

}

}

class ACCOUNT

{

Scanner sc=new Scanner(System.in);

double balance;

int amt;

long acc;

ACCOUNT(double b,long a)

{

balance=b;

acc=a;

}

double withdraw() throws Insufficient

{

System.out.println("ENTER THE AMOUNT TO BE WITHDRAWED");

amt=sc.nextInt();

if(balance>=amt)

{

balance=balance-amt;

return balance;

}

else

throw new Insufficient(balance);

}

void display()

{

System.out.println("ACCOUNT BALANCE="+balance);

}

}

class Main

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.println("ENTER THE INITIAL BALANCE");

double b=sc.nextDouble();

System.out.println("ENTER THE ACCOUNT NUMBER");

long l=sc.nextLong();

ACCOUNT obj= new ACCOUNT(b,l);

while(true)

{

System.out.println("1-WITHDRAWAL 2-DISPALY BALANCE 0-QUIT");

System.out.println("ENTER THE CHOICE");

int c=sc.nextInt();

if(c==0)

break;

switch(c)

{

case 1:

try{

obj.withdraw();

}

catch(Insufficient e)

{

System.out.println(e);

}

break;

case 2:

obj.display();

break;

default:

System.out.println("INVALID CHOICE");

}

}

}

}

