Wk10extr1 - Notepad

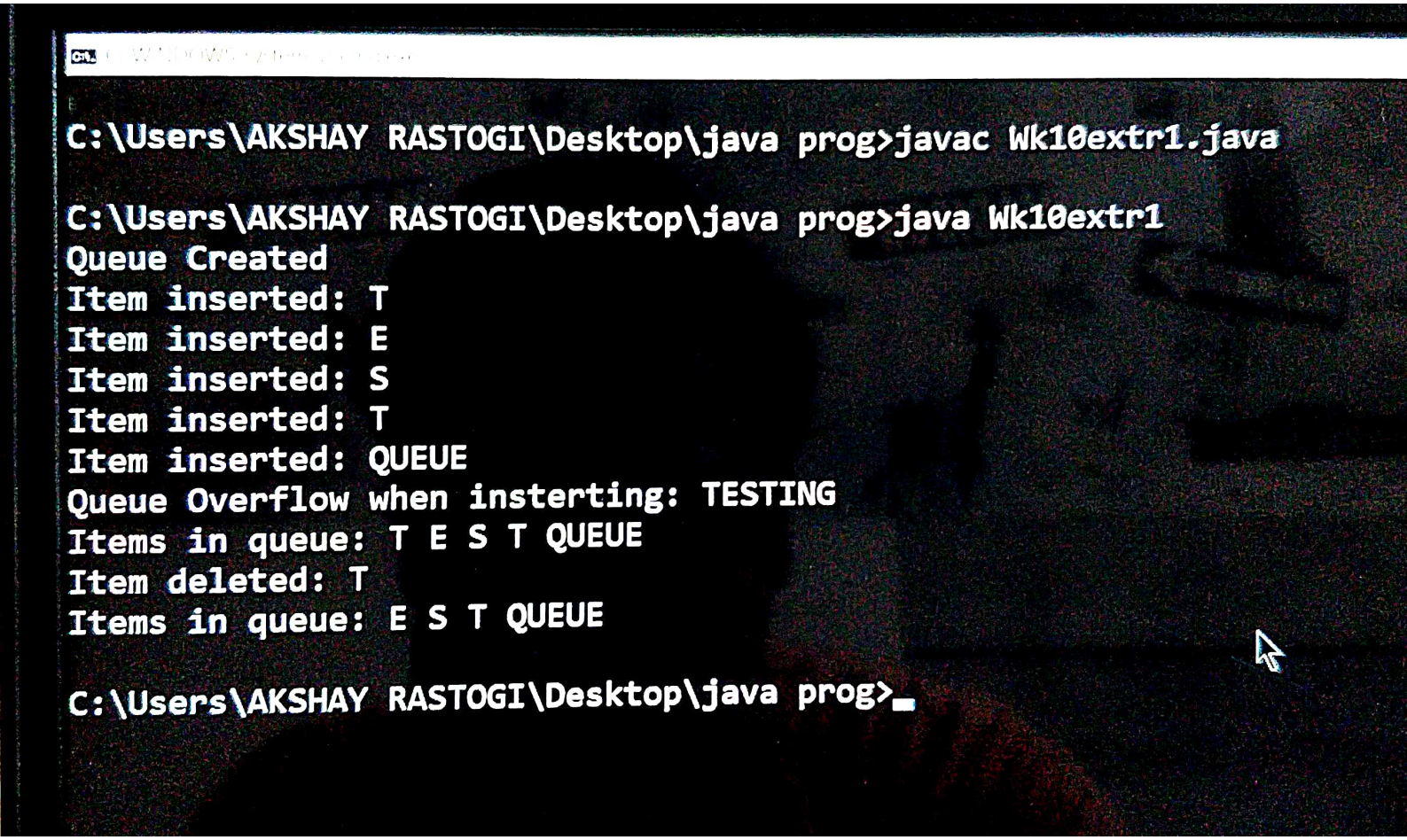
File Edit Format View Help

```
import java.util.Scanner;

interface Q<A>{
    public boolean isFull();
    public boolean isEmpty();
    public void insert(A a);
    public void delete();
    public void display();
}

@SuppressWarnings("unchecked")
class Queue<A> implements Q<A>{
    int front = 0;
    int rear = -1;
    int SIZE = 0;
    A[] arr;
    Queue(int size){
        SIZE = size;
        arr = (A[])new Object[size];
        System.out.println("Queue Created");
    }

    public boolean isFull(){
        return (rear == SIZE - 1) ? true : false;
    }

    public boolean isEmpty(){
        return (front > rear) ? true : false;

    }

    public void insert(A item){
        if(isFull()){
            System.out.println("Queue Overflow when insterting: " + item);
            return;
        }
        rear++;
        arr[rear] = item;
        System.out.println("Item inserted: " + arr[rear]);
    }

    public void delete(){
```

Wk10extr1 - Notepad

File Edit Format View Help

```
    public void delete(){
        if(isEmpty()){
            front = 0;
            rear = -1;
            System.out.println("Queue Empty");
            return;
        }
        System.out.println("Item deleted: " + arr[front++]);
    }


    public void display(){
        if(isEmpty()){
            System.out.println("Queue Empty");
            return;
        }
        System.out.print("Items in queue: ");
        for(int i = front;i<=rear;i++){
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }

}

public class Wk10extr1{
    public static void main(String[] args) {

        Queue<String> Q = new Queue<>(5);

        Q.insert("T");
        Q.insert("E");
        Q.insert("S");
        Q.insert("T");
        Q.insert("QUEUE");
        Q.insert("TESTING");
        Q.display();
        Q.delete();
        Q.display();
    }
}
```


C:\Users\AKSHAY RASTOGI\Desktop\java prog>javac Wk10extr1.java
C:\Users\AKSHAY RASTOGI\Desktop\java prog>java Wk10extr1
Queue Created
Item inserted: T
Item inserted: E
Item inserted: S
Item inserted: T
Item inserted: QUEUE
Queue Overflow when insterting: TESTING
Items in queue: T E S T QUEUE
Item deleted: T
Items in queue: E S T QUEUE
C:\Users\AKSHAY RASTOGI\Desktop\java prog>

Week 10 (Extra Programs)

1)

```
import java.util.Scanner;

interface Q<A>{
public boolean isfull();
public boolean isEmpty();
public void insert(A, a);
public void delete();
public void display();
}
@SuppressWarnings("unchecked")
class Queue<A> implements Q<A>{
int front = 0; SIZE = 0
  int rear = -1;
A[] arr ;
Queue (int size){
SIZE = size ;
arr = (A[]) new object[size];
System.out.println("Queue Created");
}
public boolean isfull(){
return (rear == SIZE-1) ? true : false ;
}}
public boolean isEmpty(){
return (front > rear) ? true : false ;
}
public void insert(A item){
if (isfull()){
System.out.println("Queue Overflow when inserting :" +item);
return ;
}
```

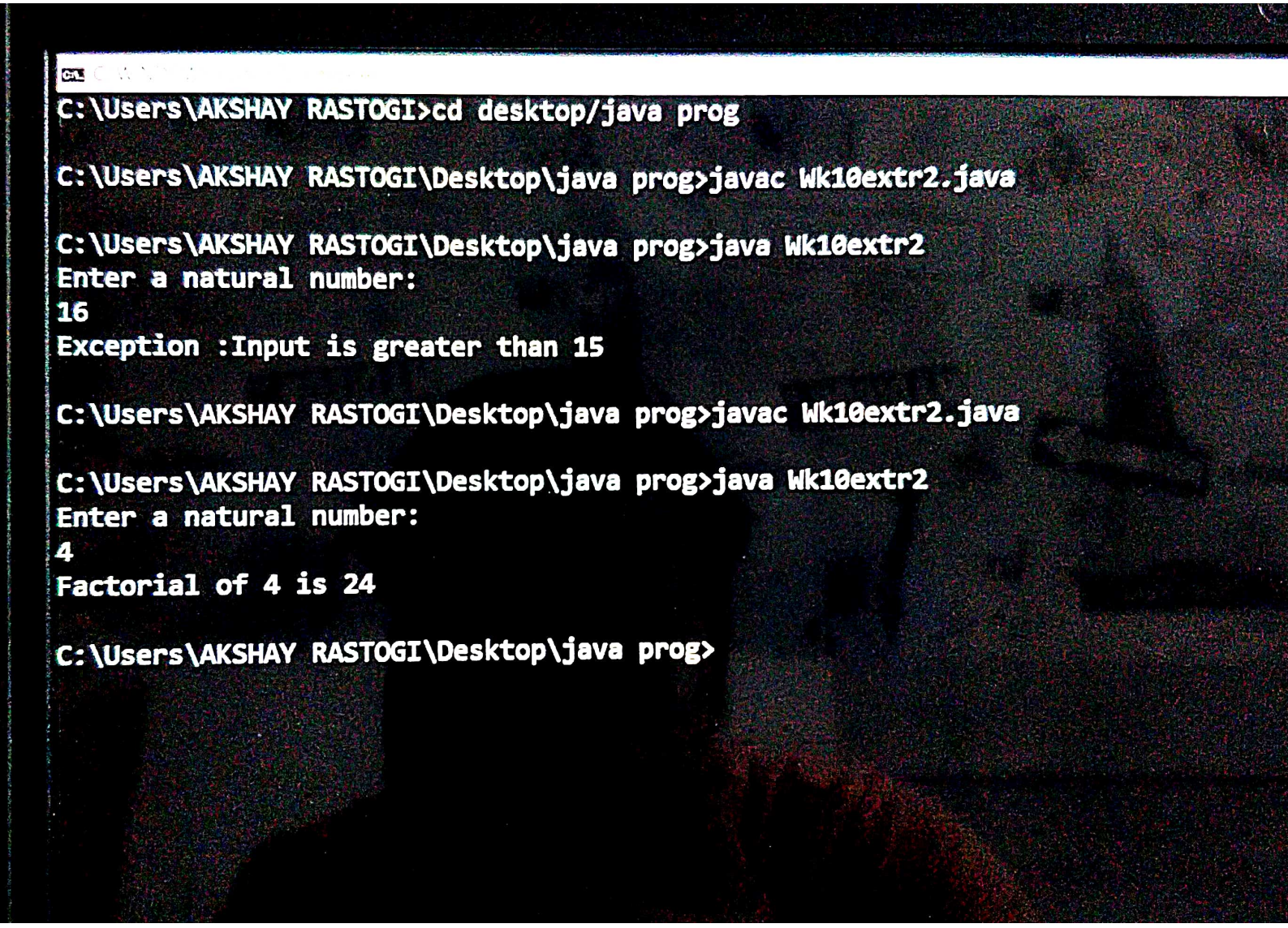
```
rear ++ ;
arr [rear] = item ;
System.out.println ("Item inserted : " + arr[rear]) ;
}
public void delete () {
if (isEmpty()) {
front = 0 ;
rear = -1 ;
System.out.println("Queue Empty") ;
return ;
}
System.out.println("Item deleted : " + arr[front++]) ;
}
public void display () {
if (isEmpty()) {
System.out.println("Queue Empty") ;
return ;
}
System.out.println("Items in Queue : ") ;
for (int i = front ; i <= rear ; i++)
System.out.println (arr[i] + " ") ;
System.out.println() ;
}
}

public class WelQueue {
public static void main (String args[]) {
Queue <String> Q = new Queue <> (5) ;
Q.insert("T") ;
Q.insert("E") ;
Q.insert ("S") ;
Q.insert("T") ;
```

```
Q. insert ("QUEUE");
Q. insert ("TESTING");
Q. display();
Q. delete();
Q. display();
}
}
```

Wk10extr2 - Notepad

File Edit Format View Help

```
import java.util.Scanner;

class myException extends Exception{
String p;
myException(String s){
p=s;
}
public String toString(){
return p;
}
}


class  Wk10extr2{

static int ComputeFactorial(int n) throws myException{
if(n > 15){
throw new myException("Input is greater than 15");
}
if(n == 0){
return 1;
}else{
return(n * ComputeFactorial(n-1));
}
}

public static void main(String[] args){
int n,fact = 0;
Scanner sc = new Scanner(System.in);
System.out.println("Enter a natural number: ");
n = sc.nextInt();
try{
fact = ComputeFactorial(n);
}catch(myException err){
System.out.println("Exception :" + err);
System.exit(1);
}
System.out.println("Factorial of " + n + " is " + fact);
}
}
```


C:\Users\AKSHAY RASTOGI>cd desktop/java prog
C:\Users\AKSHAY RASTOGI\Desktop\java prog>javac Wk10extr2.java
C:\Users\AKSHAY RASTOGI\Desktop\java prog>java Wk10extr2
Enter a natural number:
16
Exception :Input is greater than 15
C:\Users\AKSHAY RASTOGI\Desktop\java prog>javac Wk10extr2.java
C:\Users\AKSHAY RASTOGI\Desktop\java prog>java Wk10extr2
Enter a natural number:
4
Factorial of 4 is 24
C:\Users\AKSHAY RASTOGI\Desktop\java prog>

2)

```
import java.util.Scanner;

class myException extends Exception {
String p;
myException(String s) {
p = s;
}
public String toString() {
return p;
}
}

class Wk10ex2 {
static int ComputeFactorial(int n) throws myException {
if (n > 15) {
throw new myException("Input is greater than 15");
}
if (n == 0) {
return 1;
} else {
return (n * ComputeFactorial(n-1));
}
}
public static void main(String[] args) {
```

```
int n, fact = 0;
Scanner sc = new Scanner(System.in);
System.out.println("Enter a natural number: ");
n = sc.nextInt();
try {
fact = ComputeFactorial(n);
} catch (myException exp) {
System.out.println("Exception: " + exp);
System.exit(1);
}
System.out.println("Factorial of " + n + " is " + fact);
}
}
```

Wk10extr3 - Notepad

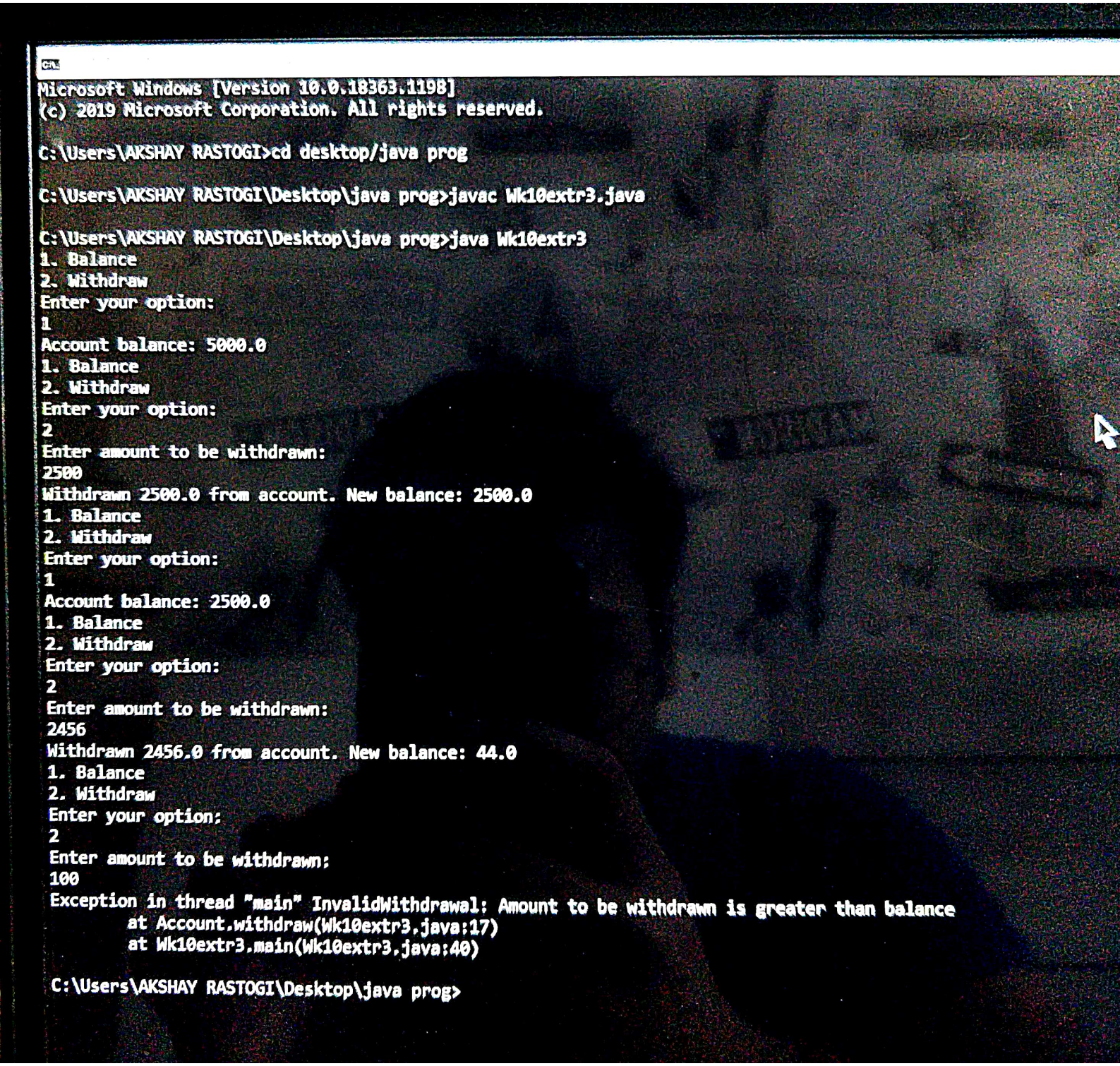
File Edit Format View Help

```
import java.util.Scanner;

class InvalidWithdrawal extends Exception{
    public InvalidWithdrawal(String s){
        super(s);
    }
}

class Account{
    double balance = 5000;
    void displayBalance(){
        System.out.println("Account balance: " + balance);
    }
    void withdraw(double amt) throws InvalidWithdrawal{
        if(amt > balance){
            throw new InvalidWithdrawal("Amount to be withdrawn is greater than balance");
        }
        balance -= amt;
        System.out.println("Withdrawn " + amt + " from account. New balance: " + balance);
    }
}

public class Wk10extr3{
    public static void main(String[] args) throws InvalidWithdrawal{
        int choice,flag = 1;
        double amount;
        Account a = new Account();
        Scanner sc = new Scanner(System.in);

        while(flag == 1){
            System.out.println("1. Balance\n2. Withdraw");
            System.out.println("Enter your option: ");
            choice = sc.nextInt();
            switch(choice){
                case 1: a.displayBalance();
                        break;
                case 2: System.out.println("Enter amount to be withdrawn: ");
                        amount = sc.nextDouble();
                        a.withdraw(amount);
                        break;
                case 3: System.exit(0);
                        break;
```

Wk10extr3 - Notepad

File Edit Format View Help

```
    }
}

class Account{
    double balance = 5000;
    void displayBalance(){
        System.out.println("Account balance: " + balance);
    }
    void withdraw(double amt) throws InvalidWithdrawal{
        if(amt > balance){
            throw new InvalidWithdrawal("Amount to be withdrawn is greater than balance");
        }
        balance -= amt;
        System.out.println("Withdrawn " + amt + " from account. New balance: " + balance);
    }
}

public class Wk10extr3{
    public static void main(String[] args) throws InvalidWithdrawal{
        int choice,flag = 1;
        double amount;
        Account a = new Account();
        Scanner sc = new Scanner(System.in);

        while(flag == 1){
            System.out.println("1. Balance\n2. Withdraw");
            System.out.println("Enter your option: ");
            choice = sc.nextInt();
            switch(choice){
                case 1: a.displayBalance();
                        break;
                case 2: System.out.println("Enter amount to be withdrawn: ");
                        amount = sc.nextDouble();
                        a.withdraw(amount);
                        break;
                case 3: System.exit(0);
                        break;
                default: System.out.println("invlaid input");
            }
        }
    }
}
```



```
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\AKSHAY RASTOGI>cd desktop/java prog

C:\Users\AKSHAY RASTOGI\Desktop\java prog>javac Wk10extr3.java

C:\Users\AKSHAY RASTOGI\Desktop\java prog>java Wk10extr3
1. Balance
2. Withdraw
Enter your option:
1
Account balance: 5000.0
1. Balance
2. Withdraw
Enter your option:
2
Enter amount to be withdrawn:
2500
Withdrawn 2500.0 from account. New balance: 2500.0
1. Balance
2. Withdraw
Enter your option:
1
Account balance: 2500.0
1. Balance
2. Withdraw
Enter your option:
2
Enter amount to be withdrawn:
2456
Withdrawn 2456.0 from account. New balance: 44.0
1. Balance
2. Withdraw
Enter your option:
2
Enter amount to be withdrawn:
100
Exception in thread "main" InvalidWithdrawal: Amount to be withdrawn is greater than balance
        at Account.withdraw(Wk10extr3.java:17)
        at Wk10extr3.main(Wk10extr3.java:40)

C:\Users\AKSHAY RASTOGI\Desktop\java prog>
```

3)
```
import java.util.Scanner;
class InvalidWithdrawal extends Exception {
public InvalidWithdrawal (String s) {
 super (s) ;
}
}

class Account {
double balance =5000 ;
void displayBalance() {
System.out.println("Account Balance = "+balance) ;
}
void withdraw (double amt) throws InvalidWithdrawal {
if (amt > balance) {
throw new InvalidWithdrawal ("Amount to be withdrawn is
greater than balance") ;
}
balane -= amt ;
```

```
System.out.println("Withdrawn "+amt+" from account. New
balance = "+balance);
}
}

public class WelcomeJava {
public static void main (String args[]) throws InvalidWithdrawal {
int choice, flag = -1;
double amount;
Account a = new Account();
Scanner sc = new Scanner (System.in);
while (flag == 1) {
System.out.println("\n1. Balance \n2. Withdraw");
System.out.println("Enter your option:");
choice = sc.nextInt();
switch (choice) {
case 1: a.displayBalance();
break;
case 2: System.out.println("Enter amount to be withdrawn:");
amount = sc.nextDouble();
a.withdrawa(amount);
break;
case 3: System.exit(0);
break;
default: System.out.println("Invalid input");
}
}
}
}
```