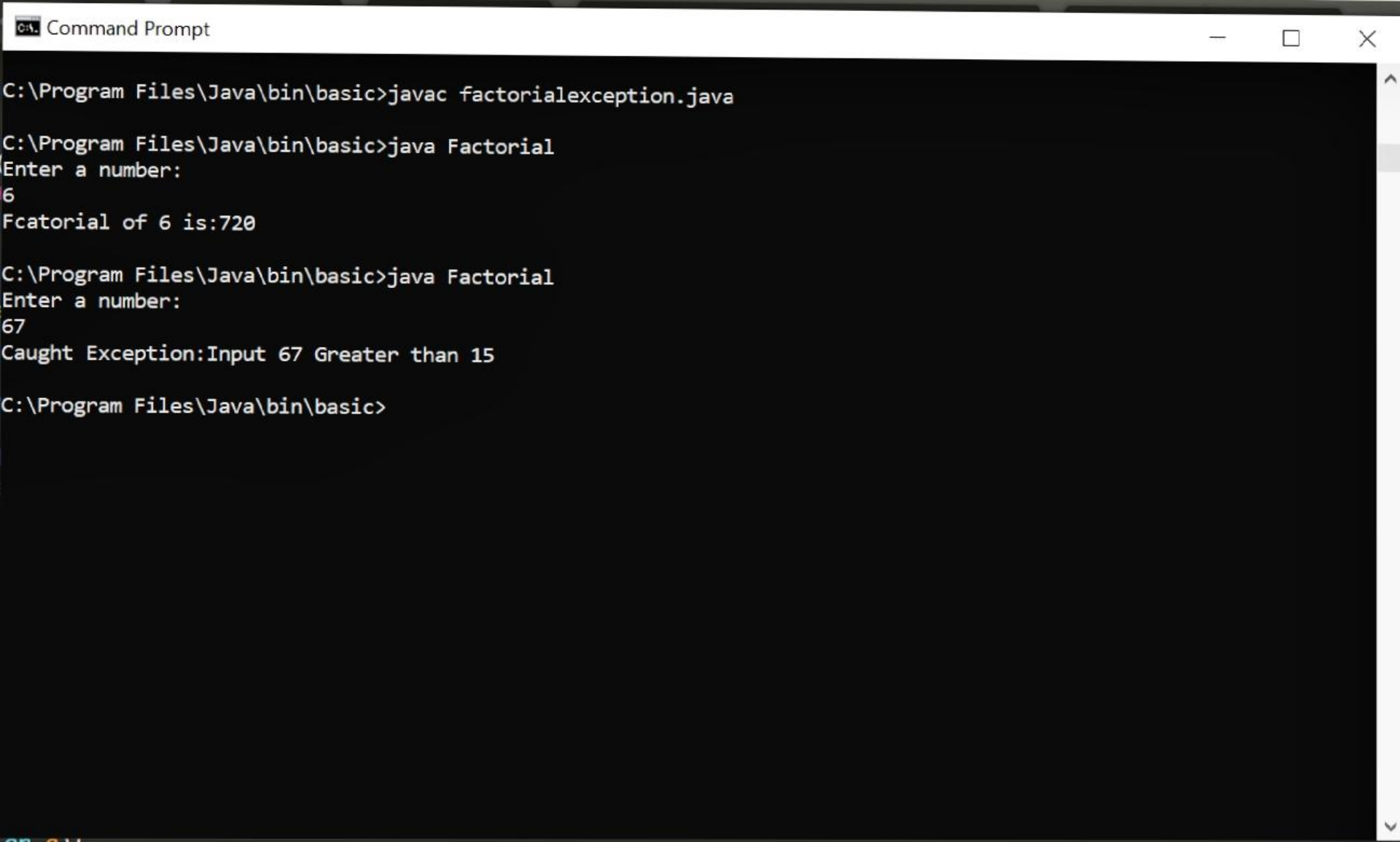
```java
/*Write a Java program to compute the factorial of a number. The input value must be tested
for validity. If it is greater than 15, the method ComputeFactorial( ) should raise an
Userdefined Exception MyException with appropriate messages.*/
import java.util.Scanner;
class MyException extends Exception{
private int detail;
MyException(int a) {
detail = a;
}
public String toString() {
return "Input "+detail+" is greater than 15";
}
}
class Factorial{
    static void ComputeFactorial(int a) throws MyException{
        int fact=1;
        if(a>15)
            throw new MyException(a);
        else{
            for(int i=1;i<=a;i++)
                fact=fact*i;
            System.out.println("Fcatorial of "+a+" is:"+fact);
        }
    }
    public static void main(String args[]){
        try{
            Scanner ss=new Scanner(System.in);
            int f;
            System.out.println("Enter a number:");
            f=ss.nextInt();
            ComputeFactorial(f);
        }
```

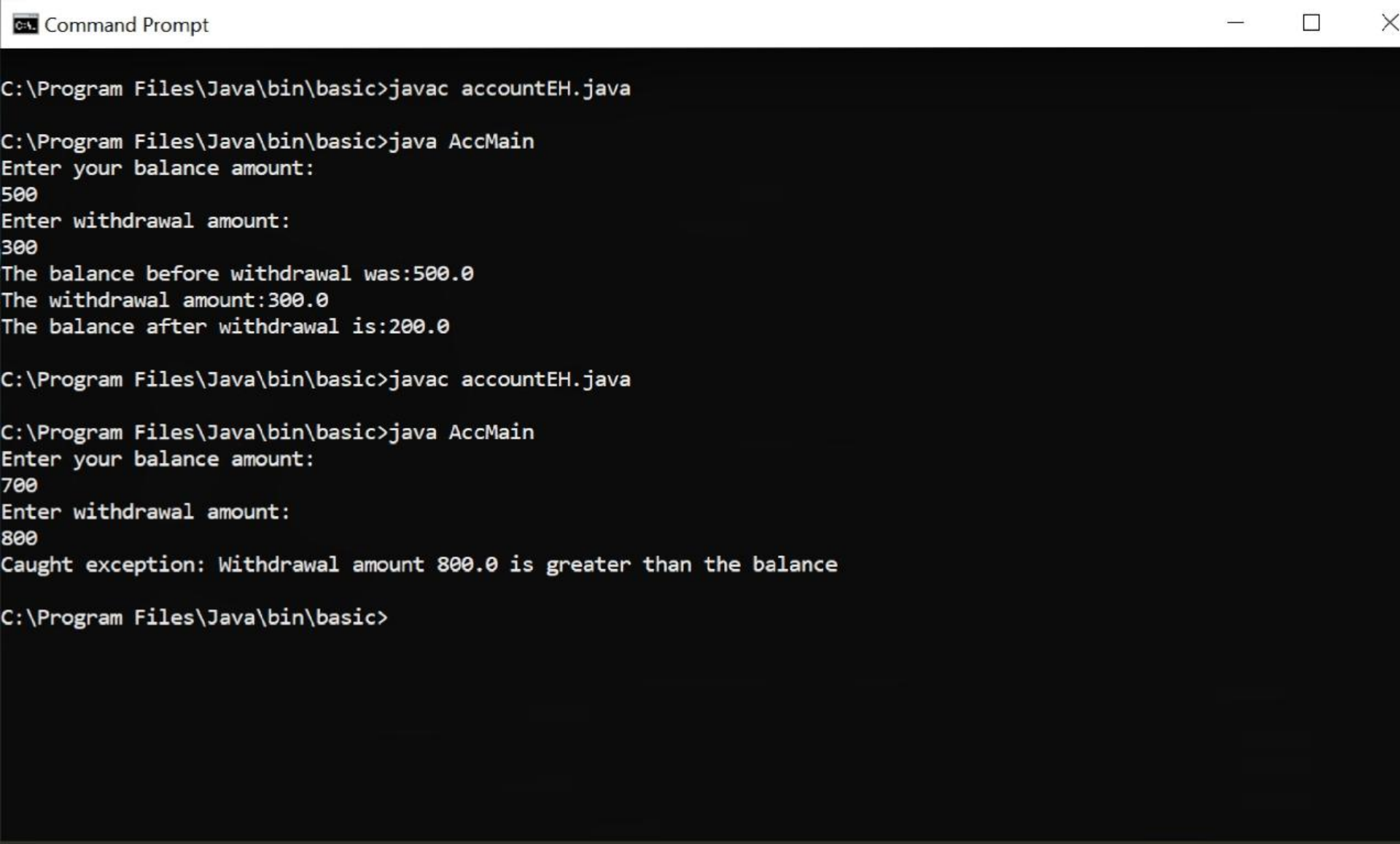
```
private int detail;
MyException(int a) {
detail = a;
}
public String toString() {
return "Input "+detail+" is greater than 15";
}
}
class Factorial{
    static void ComputeFactorial(int a) throws MyException{
        int fact=1;
        if(a>15)
            throw new MyException(a);
        else{
            for(int i=1;i<=a;i++)
                fact=fact*i;
            System.out.println("Fcatorial of "+a+" is:"+fact);
        }
    }
    public static void main(String args[]){
        try{
            Scanner ss=new Scanner(System.in);
            int f;
            System.out.println("Enter a number:");
            f=ss.nextInt();
            ComputeFactorial(f);
        }
        catch(MyException e){
            System.out.println("Caught Exception:"+e);
        }
    }
}
```


Command Prompt
C:\Program Files\Java\bin\basic>javac factorialexception.java
C:\Program Files\Java\bin\basic>java Factorial
Enter a number:
6
Fcatorial of 6 is:720
C:\Program Files\Java\bin\basic>java Factorial
Enter a number:
67
Caught Exception:Input 67 Greater than 15
C:\Program Files\Java\bin\basic>

factorialexception.java | interfaces.java | accountEH.java | test2.java | testing.java

```java
/*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.Scanner;
class MyException extends Exception{
    private double a;
    MyException(double b){
        a=b;
    }
    public String toString(){
        return "Withdrawal amount "+a+" is greater than the balance";
    }
}
class Account{
    double bal,wd;
    final double min=100.0;
    Account(){
        bal=0.0;
        wd=0.0;
    }
    void get(){
        Scanner ss=new Scanner(System.in);
        System.out.println("Enter your balance amount:");
        bal=ss.nextDouble();
        System.out.println("Enter withdrawal amount:");
        wd=ss.nextDouble();
    }
    void calc() throws MyException{
        if(wd>bal-min)
            throw new MyException(wd);
        bal-=wd;
```

```
    }
    void get(){
        Scanner ss=new Scanner(System.in);
        System.out.println("Enter your balance amount:");
        bal=ss.nextDouble();
        System.out.println("Enter withdrawal amount:");
        wd=ss.nextDouble();
    }
    void calc() throws MyException{
        if(wd>bal-min)
            throw new MyException(wd);
        bal-=wd;
    }
    void display(){
        System.out.println("The balance before withdrawal was:"+(bal+wd));
        System.out.println("The withdrawal amount:"+wd);
        System.out.println("The balance after withdrawal is:"+bal);
    }
}
class AccMain{
    public static void main(String args[]){
        try{
            Account ob=new Account();
            ob.get();
            ob.calc();
            ob.display();
        }
        catch(MyException e){
            System.out.println("Caught exception: "+e);
        }
    }
}
```


Command Prompt
C:\Program Files\Java\bin\basic>javac accountEH.java
C:\Program Files\Java\bin\basic>java AccMain
Enter your balance amount:
500
Enter withdrawal amount:
300
The balance before withdrawal was:500.0
The withdrawal amount:300.0
The balance after withdrawal is:200.0
C:\Program Files\Java\bin\basic>javac accountEH.java
C:\Program Files\Java\bin\basic>java AccMain
Enter your balance amount:
700
Enter withdrawal amount:
800
Caught exception: Withdrawal amount 800.0 is greater than the balance
C:\Program Files\Java\bin\basic>

```java
/*Implement Interfaces - QUEUE OPERATIONS*/
import java.util.Scanner;
interface Inter{
    void insert(int a);
    void delete();
    void display();
}
class Queue implements Inter{
    int q[];
    int f=0,r=-1;
    Queue(int s){
        q=new int[s];
    }
    public void insert(int item){
        if(r==(q.length-1)){
            System.out.println("Queue is full");
        }
        else{
            r++;
            q[r]=item;
        }
    }
    public void delete(){
        if(f>r){
            System.out.println("Queue is empty");
            f=0;
            r=-1;
        }
        else
            f++;
    }
    public void display(){
        if(f>r){
            System.out.println("Queue is empty");
        }
        else{
        System.out.println("The contents of the queue:");
        for(int i=f;i<=r;i++){
            System.out.println(q[i]);
```

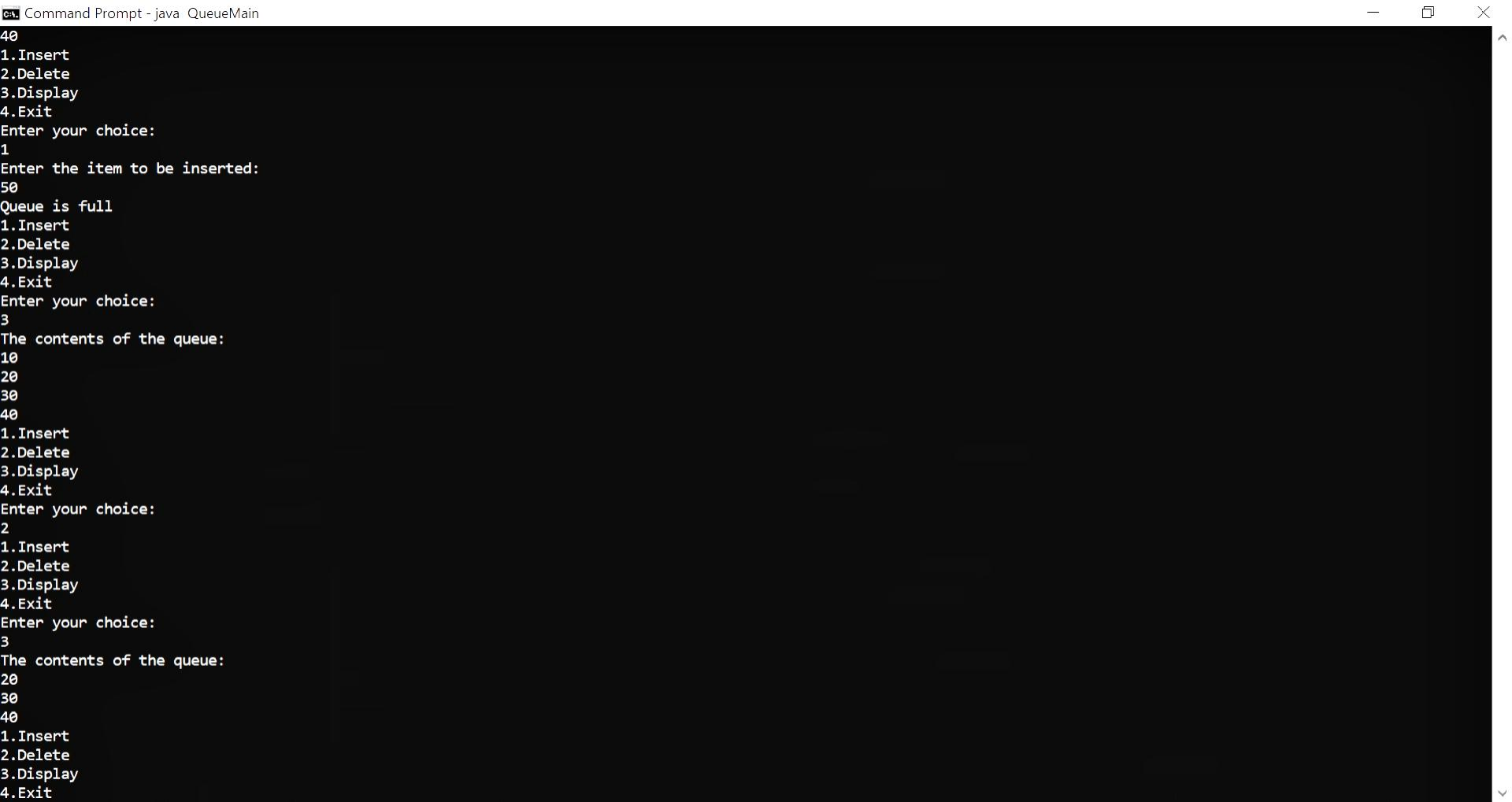
```
27      public void display(){
28          if(f>r){
29              System.out.println("Queue is empty");
30          }
31          else{
32          System.out.println("The contents of the queue:");
33          for(int i=f;i<=r;i++){
34              System.out.println(q[i]);
35          }
36      }
37      }
38  }
39  class QueueMain{
40      public static void main(String args[]){
41          Scanner ss=new Scanner(System.in);
42          Queue ob=new Queue();
43          for(;;){
44              System.out.println("1.Insert\n2.Delete\n3.Display\n4.Exit");
45              System.out.println("Enter your choice:");
46              int choice=ss.nextInt();
47          switch(choice){
48              case 1:{
49                  System.out.println("Enter the item to be inserted:");
50                  int item=ss.nextInt();
51                  ob.insert(item);
52                  break;
53              }
54              case 2:{
55                  ob.delete();
56                  break;
57              }
58              case 3:{
```

```java
class QueueMain{
    public static void main(String args[]){
        Scanner ss=new Scanner(System.in);
        Queue ob=new Queue();
        for(;;){
            System.out.println("1.Insert\n2.Delete\n3.Display\n4.Exit");
            System.out.println("Enter your choice:");
            int choice=ss.nextInt();
        switch(choice){
            case 1:{
                System.out.println("Enter the item to be inserted:");
                int item=ss.nextInt();
                ob.insert(item);
                break;
            }
            case 2:{
                ob.delete();
                break;
            }
            case 3:{
                ob.display();
                break;
             }
             default:{
                System.out.println("Wrong input");
                break;
             }
        }
    }
        }
}
```

Command Prompt - java QueueMain

```
C:\Program Files\Java\bin\basic>javac interfaces.java

C:\Program Files\Java\bin\basic>java QueueMain
Enter the size of the queue:
4
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
1
Enter the item to be inserted:
10
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
1
Enter the item to be inserted:
20
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
1
Enter the item to be inserted:
30
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
1
Enter the item to be inserted:
40
1.Insert
2.Delete
3.Display
```


Command Prompt - java QueueMain
40
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
1
Enter the item to be inserted:
50
Queue is full
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
3
The contents of the queue:
10
20
30
40
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
2
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
3
The contents of the queue:
20
30
40
1.Insert
2.Delete
3.Display
4.Exit

Command Prompt - java QueueMain

```
The contents of the queue:
20
30
40
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
2
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
3
The contents of the queue:
30
40
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
2
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
3
The contents of the queue:
40
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
2
1.Insert
2.Delete
```


Command Prompt - java QueueMain
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
2
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
3
The contents of the queue:
40
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
2
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
3
Queue is empty
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:
4
Wrong input
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice: