LAB-9

MAJJIGA JASWANTH

20BCD7171

Q 1a)

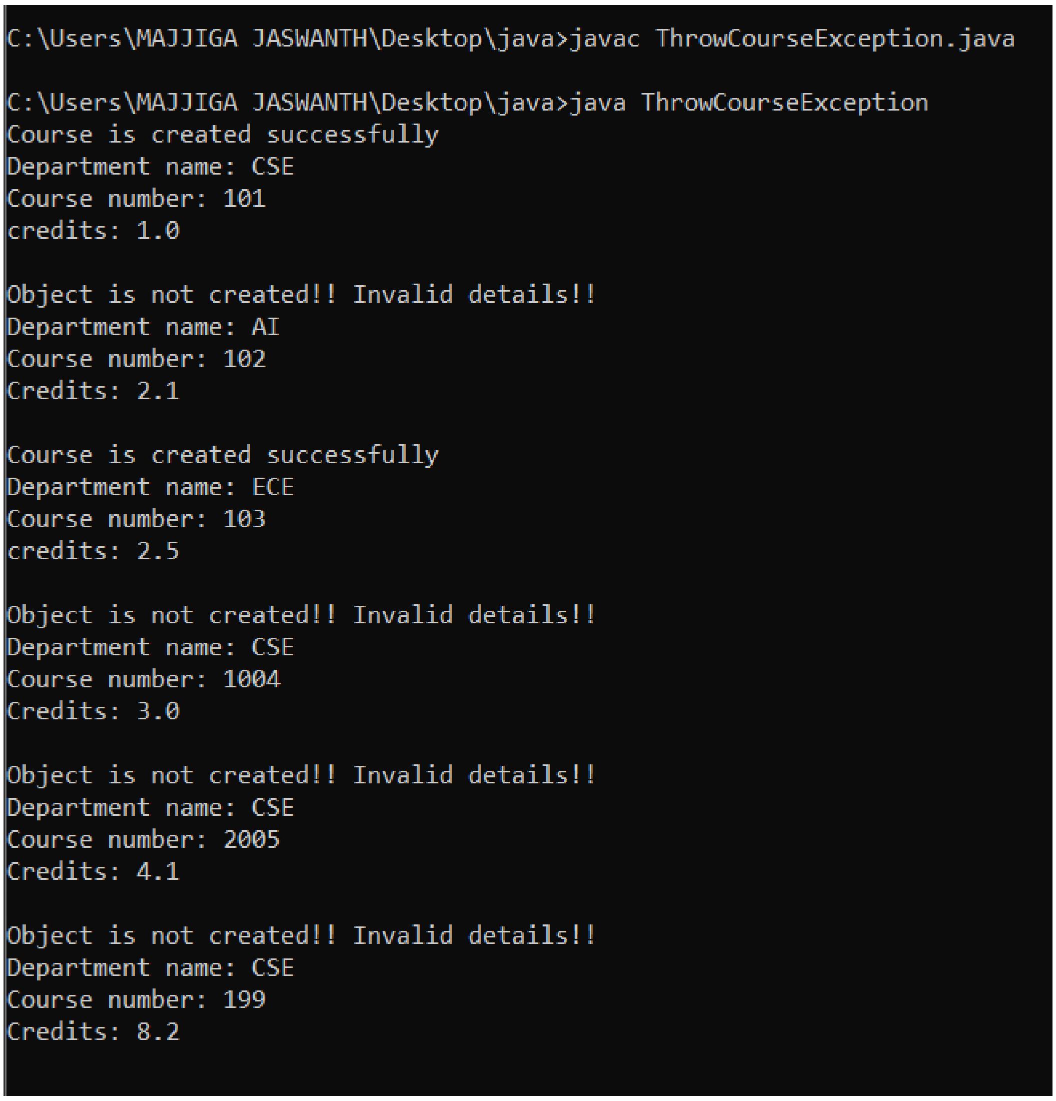
Create a Checked CourseException class and whose constructor receives a String that holds a college course's department (for example, CIS), a course number (for example, 101), and a number of credits (for example, 3). Save the file as CourseException.java. Create a Course class with the same fields and whose constructor requires values for each field. Upon construction, throw a CourseException if the department does not consist of three letters, if the course number does not consist of three digits between 100 and 499 inclusive, or if the credits are less than 1 or more than 6. Save the class as Course.java. Write an application that establishes an array of at least six Course objects with valid and invalid values. Display an appropriate message when a Course object is created successfully and when one is not. Save the file as ThrowCourseException.java.

CODE:

```
import java.util.*;

public class ThrowCourseException

{

public static void main(String[] args)throws CourseException

{

course c[] = new course[6];

c[0]=new course("CSE", 101, 1.0);

c[1]=new course("AI", 102, 2.1);

c[2]=new course("ECE", 103, 2.5);

c[3]=new course("CSE", 1004, 3.0);

c[4]=new course("CSE", 2005, 4.1);

c[5]=new course("CSE",199,8.2);

}

}


class course{

private String department;

private int courseNumber;
private double credits;

public course(String department,int courseNumber,double credits)throws
CourseException

{

if(department.length()!=3||(courseNumber<100||
courseNumber>499)||(credits<1||

credits>6))

{

try

{

throw new CourseException(department,courseNumber,credits);

}

catch(CourseException ex)

{

}

}

else

{

System.out.println("Course is created successfully \nDepartment name:
"+department+"\nCourse number: "+courseNumber+"\ncredits:
"+credits+"\n" );

}

this.department = department;

this.courseNumber = courseNumber;

this.credits = credits;

}

}


 class CourseException extends Exception{

public CourseException(String dept,int course,double cred){

System.out.println("Object is not created!! Invalid
details!!"+"\nDepartment name: "+dept+"\nCourse number:
"+course+"\nCredits: "+cred+"\n");

}
```

```
}
```

OUTPUT:



```
C:\Users\MAJJIGA JASWANTH\Desktop\java>javac ThrowCourseException.java

C:\Users\MAJJIGA JASWANTH\Desktop\java>java ThrowCourseException
Course is created successfully
Department name: CSE
Course number: 101
credits: 1.0

Object is not created!! Invalid details!!
Department name: AI
Course number: 102
Credits: 2.1

Course is created successfully
Department name: ECE
Course number: 103
credits: 2.5

Object is not created!! Invalid details!!
Department name: CSE
Course number: 1004
Credits: 3.0

Object is not created!! Invalid details!!
Department name: CSE
Course number: 2005
Credits: 4.1

Object is not created!! Invalid details!!
Department name: CSE
Course number: 199
Credits: 8.2
```

Q-1b)

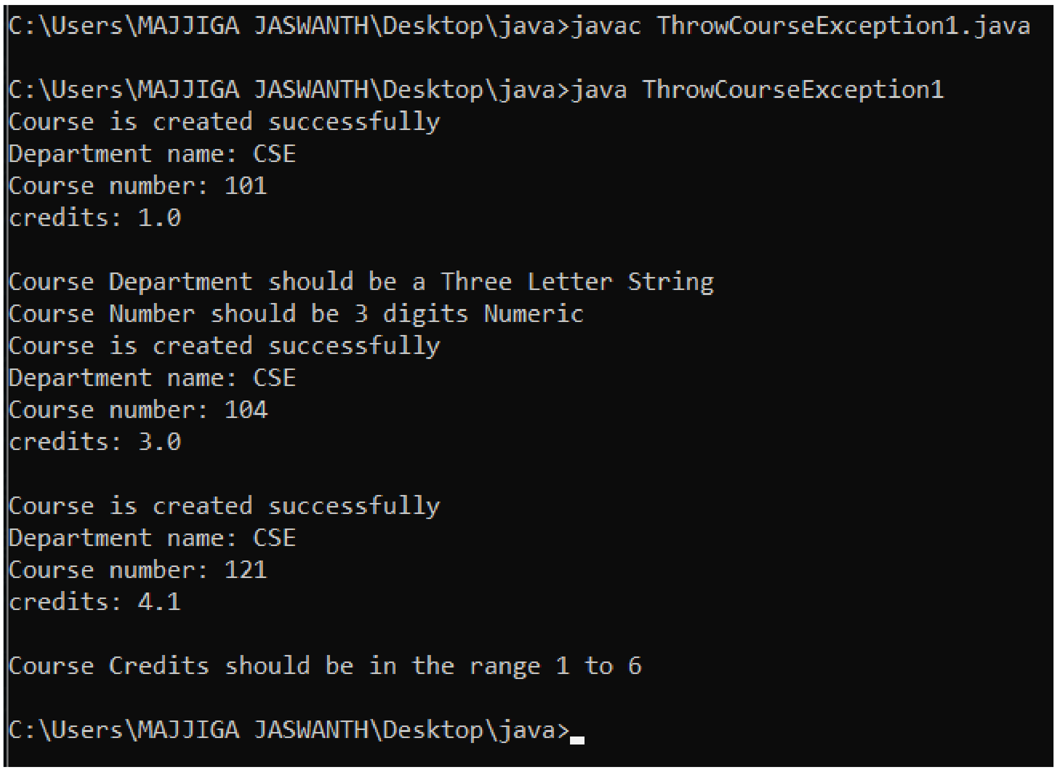
1. b. Now modify the above *CourseException* class to display different message based on the cause of the exception as shown below:

Different Message to be displayed based on the cause, while exception is caught
Course Department should be a Three Letter String
Course Number should be 3 digits Numeric
Course Credits should be in the range 1 to 6.

CODE:

```
import java.util.*;

public class ThrowCourseException1

{

public static void main(String[] args)throws CourseException

{

course c[] = new course[6];

c[0]=new course("CSE", 101, 1.0);

c[1]=new course("JAVA", 102, 2.1);

c[2]=new course("ECE", 1003, 2.5);

c[3]=new course("CSE", 104, 3.0);

c[4]=new course("CSE", 121, 4.1);

c[5]=new course("CSE",199,8.2);

}

}

 class course{

private String department;

private int courseNumber;

private double credits;

public course(String department,int courseNumber,double credits)throws
CourseException

{

if(department.length()!=3||(courseNumber<100||
courseNumber>499)||(credits<1||

credits>6))

{

try

{

throw new CourseException(department,courseNumber,credits);

}

catch(CourseException ex)

{

}
}

else

{

System.out.println("Course is created successfully \nDepartment name: 
"+department+"\nCourse number: "+courseNumber+"\ncredits: 
"+credits+"\n" );

}

this.department = department;

this.courseNumber = courseNumber;

this.credits = credits;

}

}


 class CourseException extends Exception{

public CourseException(String dept,int course,double cred){

 if(dept.length()!=3){

 System.out.println("Course Department should be a Three Letter 
String");

 }

 if(course<100|| course>499){

 System.out.println("Course Number should be 3 digits Numeric");

 }

 if(cred<1|| cred>6){

 System.out.println("Course Credits should be in the range 1 to 6");

 }

}

}
```

OUTPUT:



```
C:\Users\MAJJIGA JASWANTH\Desktop\java>javac ThrowCourseException1.java

C:\Users\MAJJIGA JASWANTH\Desktop\java>java ThrowCourseException1
Course is created successfully
Department name: CSE
Course number: 101
credits: 1.0

Course Department should be a Three Letter String
Course Number should be 3 digits Numeric
Course is created successfully
Department name: CSE
Course number: 104
credits: 3.0

Course is created successfully
Department name: CSE
Course number: 121
credits: 4.1

Course Credits should be in the range 1 to 6

C:\Users\MAJJIGA JASWANTH\Desktop\java>
```

Q

2. a. Develop a **ScoreGradeTest** class to display a series of five Student ID numbers and asks the user to enter a numeric test scores of three subjects and a grade. Develop a checked *ScoreException* class and throw it when the user enters an invalid score (greater than 100 and less than 0). Develop a checked *GradeException* class and throw it when the user enters an invalid grade. (Other than A, B, C, D, F). Also throw *GradeException* if the user enters wrong grade for their average test score. Differentiate these two GradeException types with appropriate messages.

Average Test Score	**Grade**
>90	A
>80 and <89	B
>70 and <79	C
>60 and <69	D
<60	F

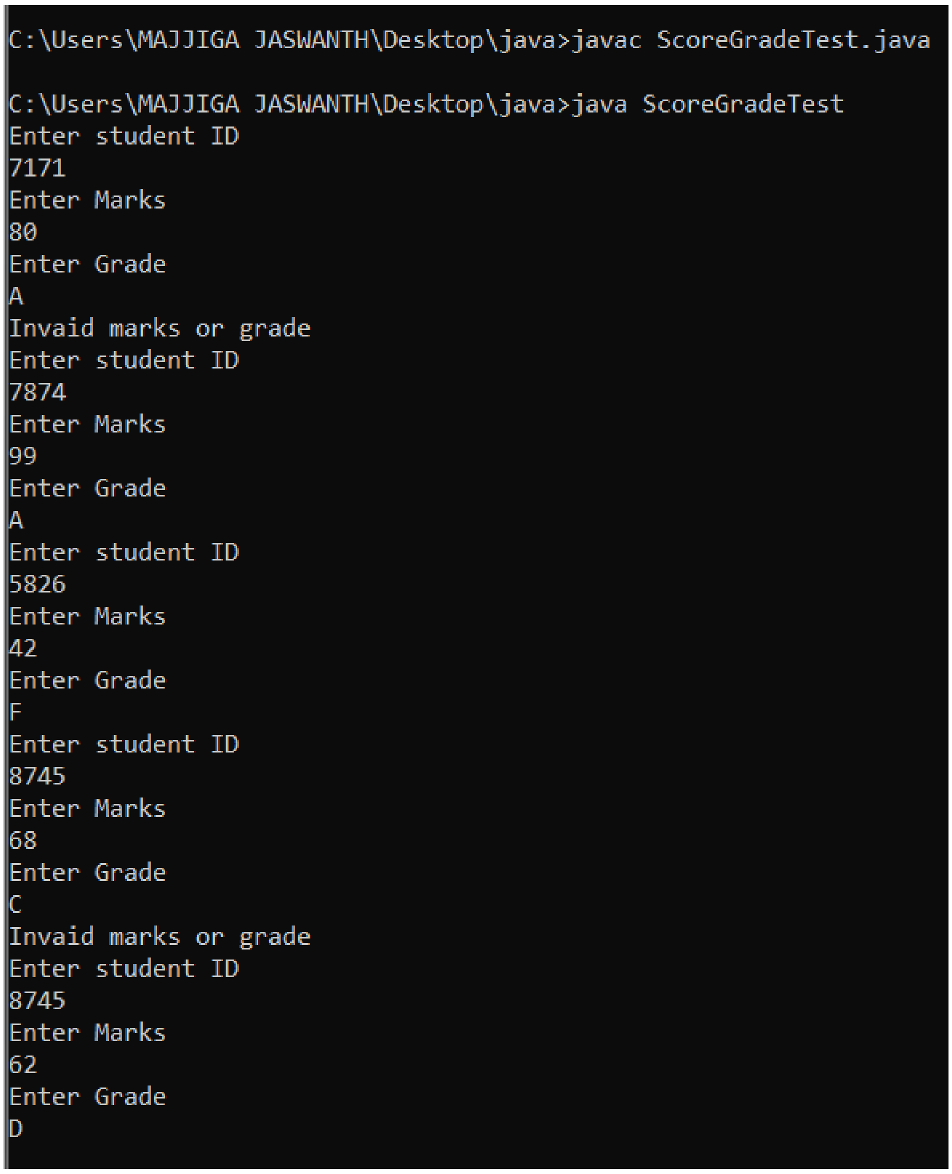
CODE:

```
import java.util.*;
public class ScoreGradeTest
{
public static void main(String[] args) {
```

```
Scanner sc=new Scanner(System.in);
for(int i=1;i<=5;i++){
System.out.println("Enter student ID");
int sid=sc.nextInt();
System.out.println("Enter Marks");
int marks=sc.nextInt();
System.out.println("Enter Grade");
String grade=sc.next();
String gr="";
scoredexception s=new scoredexception();
gradeexception g=new gradeexception();
if(marks>90){
gr="A";
}
else if(marks>80 && marks<89){
gr="B";
}
else if(marks>70 && marks<79){
gr="C";
}
else if(marks>60 && marks<69){
gr="D";
}
else if(marks<60){
gr="F";
}
try{
if(marks>100){
s.score();
}
}
catch(Exception e){
```

```
System.out.println("Invaid marks or grade");
}
boolean x=grade.equals(gr);
try{
if(x==false){
s.score();
}
}
catch(Exception e){
System.out.println("Invaid marks or grade");
}
}
}
}
class scoredexception{
void score ()throws Exception{
throw new Exception("invalid marks");
}
}
class gradeexception{
void grade() throws Exception{
throw new Exception("invalid grade");
}
}
```

OUTPUT:



```
C:\Users\MAJJIGA JASWANTH\Desktop\java>javac ScoreGradeTest.java

C:\Users\MAJJIGA JASWANTH\Desktop\java>java ScoreGradeTest
Enter student ID
7171
Enter Marks
80
Enter Grade
A
Invaid marks or grade
Enter student ID
7874
Enter Marks
99
Enter Grade
A
Enter student ID
5826
Enter Marks
42
Enter Grade
F
Enter student ID
8745
Enter Marks
68
Enter Grade
C
Invaid marks or grade
Enter student ID
8745
Enter Marks
62
Enter Grade
D
```

Q-2b)

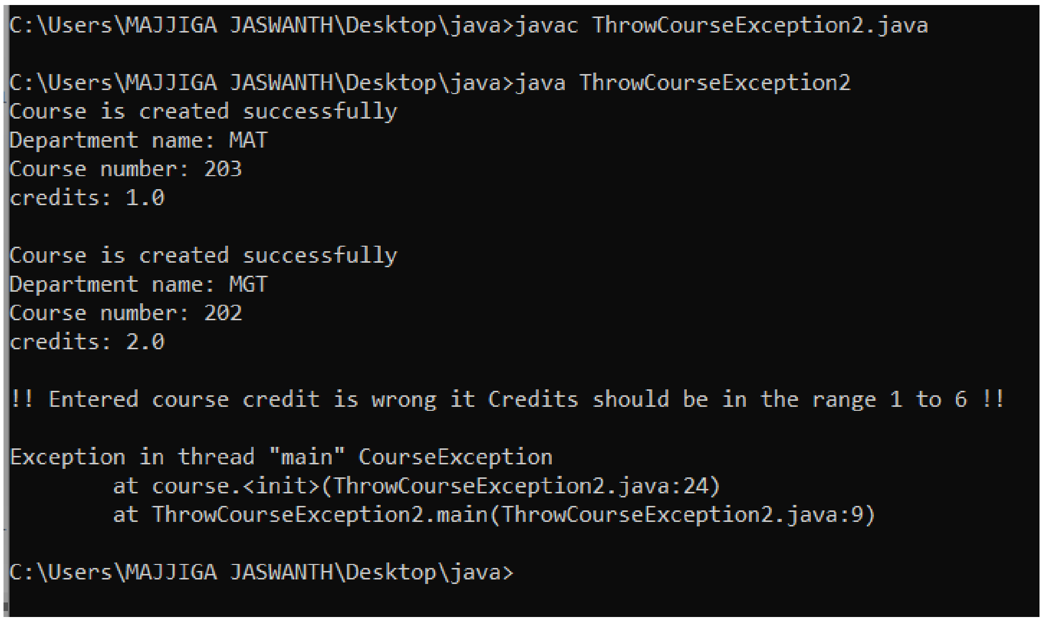
Convert Checked exceptions developed in (1b) and (2a) questions into Un-Checked exceptions. Write appropriateobservationsand differencesthat you identifiedafterconversion.

CODE:

```
import java.util.*;

public class ThrowCourseException2

{

public static void main(String[] args)throws CourseException

{

course c[] = new course[6];
c[0]=new course("MAT",203,1);

c[1]=new course("MGT",202,2);

c[2]=new course("MAT",301,7);

c[3]=new course("STS",302,4);

c[4]=new course("CSE",2005,5);

c[5]=new course("PHY",102,6);

}

}

 class course{

private String department;

private int courseNumber;

private double credits;

public course(String department,int courseNumber,double credits)throws
CourseException

{

if(department.length()!=3||(courseNumber<100||
courseNumber>499)||(credits<1||

credits>6))

{

throw new CourseException(department,courseNumber,credits);

}

else

{

System.out.println("Course is created successfully \nDepartment name:

"+department+"\nCourse number: "+courseNumber+"\ncredits:
"+credits+"\n" );

}

this.department = department;

this.courseNumber = courseNumber;

this.credits = credits;

}

}
```

```
 class CourseException extends Exception{

public CourseException(String dept,int course,double cred){

 if(dept.length()!=3){

 System.out.println("!! Entered course is wrong it should be a Three
Letter String !!");

 System.out.println();

 }

 if(course<100|| course>499){

 System.out.println("!! Entered course number is wrong it should be 3
digits Numeric

!!");

 System.out.println();

 }

 if(cred<1|| cred>6){

 System.out.println("!! Entered course credit is wrong it Credits
should be in the range 1

to 6 !!");

 System.out.println();

 }

}

}
```

OUTPUT:



```
C:\Users\MAJJIGA JASWANTH\Desktop\java>javac ThrowCourseException2.java

C:\Users\MAJJIGA JASWANTH\Desktop\java>java ThrowCourseException2
Course is created successfully
Department name: MAT
Course number: 203
credits: 1.0

Course is created successfully
Department name: MGT
Course number: 202
credits: 2.0

!! Entered course credit is wrong it Credits should be in the range 1 to 6 !!

Exception in thread "main" CourseException
        at course.<init>(ThrowCourseException2.java:24)
        at ThrowCourseException2.main(ThrowCourseException2.java:9)

C:\Users\MAJJIGA JASWANTH\Desktop\java>
```

CODE FOR UNCHECKED EXCEPTION:

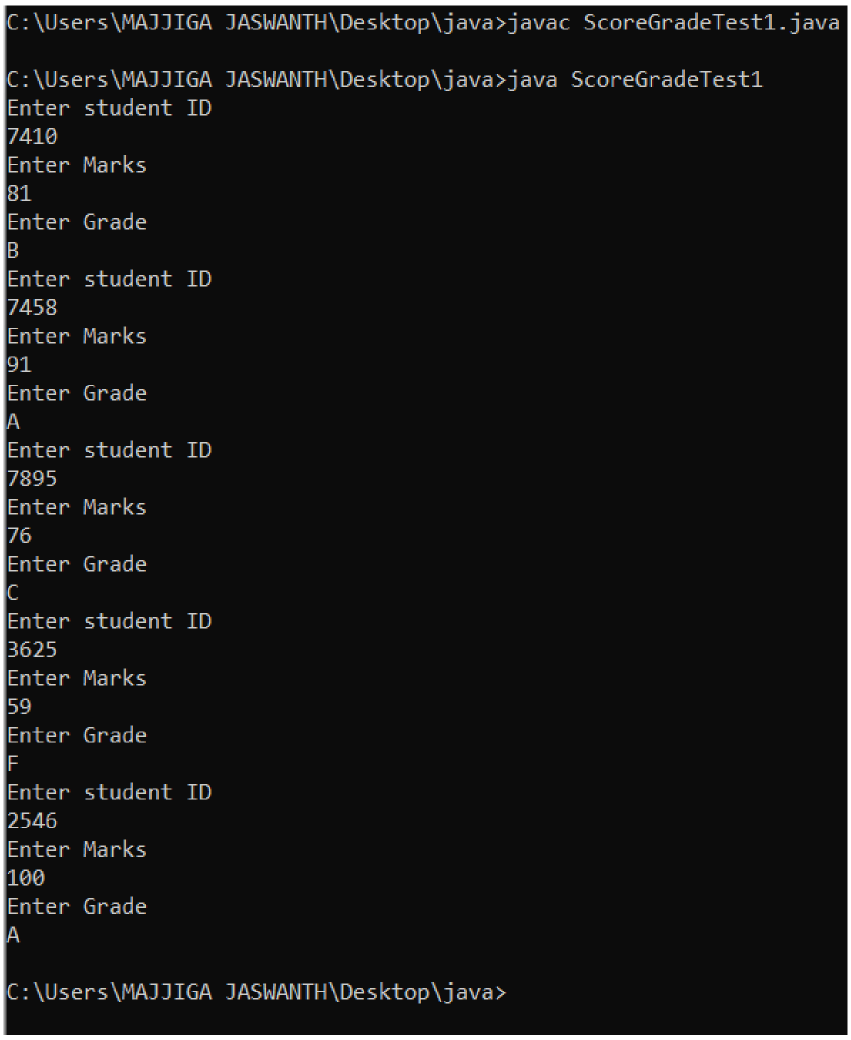
2a)

Code:

```java
import java.util.*;
public class ScoreGradeTest1
{
public static void main(String[] args) throws Exception {
Scanner sc=new Scanner(System.in);
for(int i=1;i<=5;i++){
System.out.println("Enter student ID");
int sid=sc.nextInt();
System.out.println("Enter Marks");
int marks=sc.nextInt();
System.out.println("Enter Grade");
String grade=sc.next();
String gr="";
if(marks>90){
gr="A";
}
else if(marks>80 && marks<89){
gr="B";
}
else if(marks>70 && marks<79){
gr="C";
}
else if(marks>60 && marks<69){
gr="D";
}
else if(marks<60){
gr="F";
}
if(marks>100){
```

```
throw new scoredexception();
}
boolean x=grade.equals(gr);
if(x==false){
throw new gradeexception();
}
}
}
}
class scoredexception extends Exception{
 public scoredexception (){
 System.out.println("invalid marks");
}
}
class gradeexception extends Exception{
void gradeexception() {
System.out.println("invalid grade");
}
}
```

Output:



```
C:\Users\MAJJIGA JASWANTH\Desktop\java>javac ScoreGradeTest1.java

C:\Users\MAJJIGA JASWANTH\Desktop\java>java ScoreGradeTest1
Enter student ID
7410
Enter Marks
81
Enter Grade
B
Enter student ID
7458
Enter Marks
91
Enter Grade
A
Enter student ID
7895
Enter Marks
76
Enter Grade
C
Enter student ID
3625
Enter Marks
59
Enter Grade
F
Enter student ID
2546
Enter Marks
100
Enter Grade
A

C:\Users\MAJJIGA JASWANTH\Desktop\java>
```

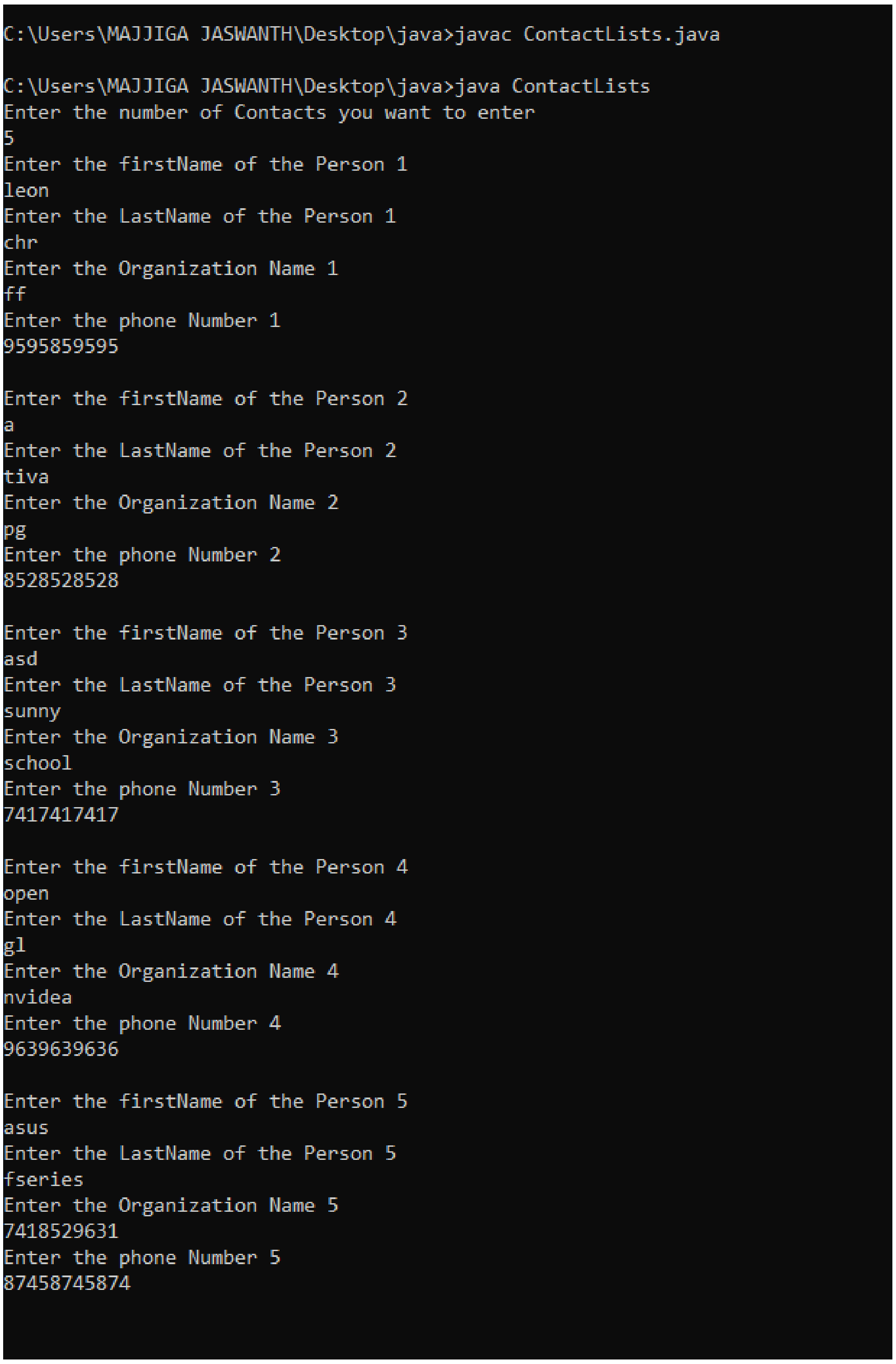
Q-3)

Develop a class “Contacts” with attributes first name, last name, organization, mobile number. Develop another class “ContactList” to hold all contacts of a user in the form of Contacts class objectsand as anArray List.Develop a menu-basedapplication to add, to delete, to update, and to display all contacts. Duringdisplayof each contactobject from the ArrayList the developer not able to access attributes. Find the way to display to all objects using Java predefined methods,inthe order of insertions taken place into anArrayList.

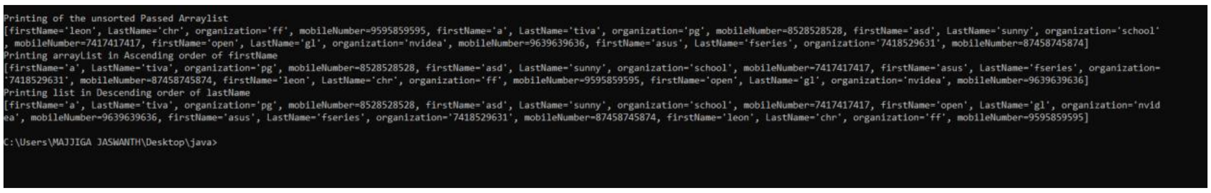
Code:

```
import java.util.Comparator;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;
import java.util.Scanner;

public class ContactLists {

 public static void main(String[] args) {

 String firstName;

 String lastName;

 String organization;

 long mobileNumber;

 Scanner scanner = new Scanner(System.in);

 ArrayList<Contacts> contactListArray = new ArrayList<>();

 System.out.println("Enter the number of Contacts you want to enter");

 int n = scanner.nextInt();

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

 {

 System.out.println("Enter the firstName of the Person "+i);

 firstName=scanner.next();

 System.out.println("Enter the LastName of the Person "+i);

 lastName = scanner.next();

 System.out.println("Enter the Organization Name "+i);

 organization = scanner.next();

 System.out.println("Enter the phone Number "+i);

 mobileNumber = scanner.nextLong();

 contactListArray.add(new
Contacts(firstName,lastName,organization,mobileNumber));

 System.out.println();

 }

 System.out.println("Printing of the unsorted Passed Arraylist");

 System.out.println(contactListArray);

 System.out.println("Printing arrayList in Ascending order of
firstName");

 Collections.sort(contactListArray, new FirstNameSorter());

 System.out.println(contactListArray);

 System.out.println("Printing list in Descending order of lastName");
 Comparator<Contacts> comparator = Comparator.comparing(e ->
e.getLastName());
 contactListArray.sort(comparator.reversed());
 System.out.println(contactListArray);
}
}
 class Contacts{
 String firstName;
 String LastName;
 String organization;
 long mobileNumber;
 public Contacts(String firstName, String lastName, String
organization, long
mobileNumber) { // constrcutor of contact class
 this.firstName = firstName;
 this.LastName = lastName;
 this.organization = organization;
 this.mobileNumber = mobileNumber;
 }
 public String getFirstName() {
 return firstName;
 }
 public String getLastName() {
 return LastName;
 }
 public String getOrganization() {
 return organization;
 }
 public long getMobileNumber() {
 return mobileNumber;
 }
 @Override
 public String toString() {
 return
 "firstName='" + firstName + '\'' +
 ", LastName='" + LastName + '\'' +
 ", organization='" + organization + '\'' +
 ", mobileNumber=" + mobileNumber;
 }
}
class FirstNameSorter implements Comparator<Contacts>{
 @Override
 public int compare(Contacts o1, Contacts o2) {
 return o1.getFirstName().compareTo(o2.getFirstName());
 }
}
```

Output:



```
C:\Users\MAJJIGA JASWANTH\Desktop\java>javac ContactLists.java

C:\Users\MAJJIGA JASWANTH\Desktop\java>java ContactLists
Enter the number of Contacts you want to enter
5
Enter the firstName of the Person 1
leon
Enter the LastName of the Person 1
chr
Enter the Organization Name 1
ff
Enter the phone Number 1
9595859595

Enter the firstName of the Person 2
a
Enter the LastName of the Person 2
tiva
Enter the Organization Name 2
pg
Enter the phone Number 2
8528528528

Enter the firstName of the Person 3
asd
Enter the LastName of the Person 3
sunny
Enter the Organization Name 3
school
Enter the phone Number 3
7417417417

Enter the firstName of the Person 4
open
Enter the LastName of the Person 4
gl
Enter the Organization Name 4
nvidea
Enter the phone Number 4
9639639636

Enter the firstName of the Person 5
asus
Enter the LastName of the Person 5
fseries
Enter the Organization Name 5
7418529631
Enter the phone Number 5
87458745874
```



```
Printing of the unsorted Passed Arraylist
[firstName='leon', LastName='chr', organization='ff', mobileNumber=9595859595, firstName='a', LastName='tiva', organization='pg', mobileNumber=8528528528, firstName='asd', LastName='sunny', organization='school'
, mobileNumber=7417417417, firstName='open', LastName='gl', organization='nvidea', mobileNumber=9639639636, firstName='asus', LastName='fseries', organization='7418529631', mobileNumber=87458745874]
Printing arrayList in Ascending order of firstName
[firstName='a', LastName='tiva', organization='pg', mobileNumber=8528528528, firstName='asd', LastName='sunny', organization='school', mobileNumber=7417417417, firstName='asus', LastName='fseries', organization=
'7418529631', mobileNumber=87458745874, firstName='leon', LastName='chr', organization='ff', mobileNumber=9595859595, firstName='open', LastName='gl', organization='nvidea', mobileNumber=9639639636]
Printing list in Descending order of lastName
[firstName='a', LastName='tiva', organization='pg', mobileNumber=8528528528, firstName='asd', LastName='sunny', organization='school', mobileNumber=7417417417, firstName='open', LastName='gl', organization='nvid
ea', mobileNumber=9639639636, firstName='asus', LastName='fseries', organization='7418529631', mobileNumber=87458745874, firstName='leon', LastName='chr', organization='ff', mobileNumber=9595859595]

C:\Users\MAJJIGA JASWANTH\Desktop\java>
```