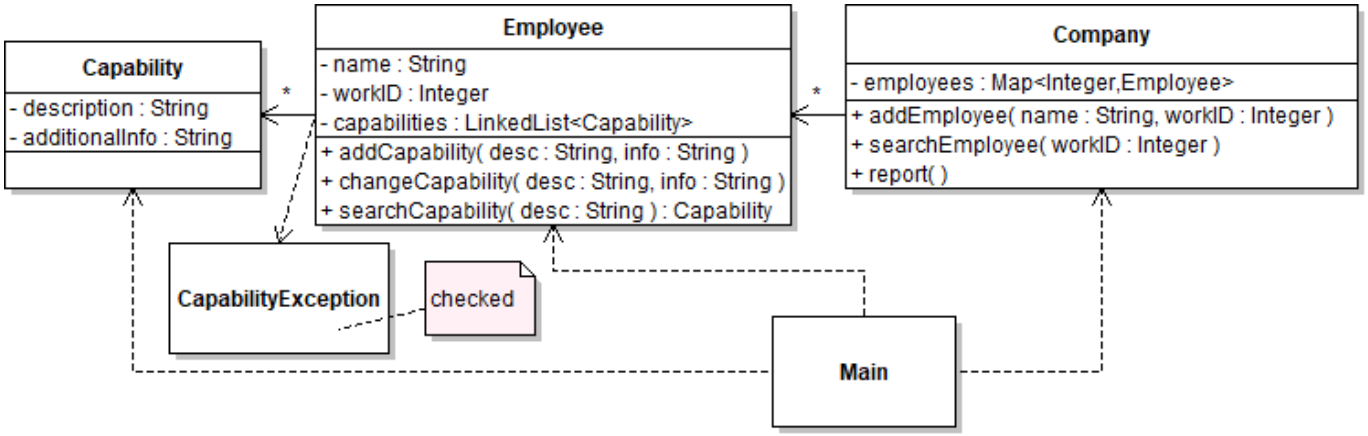
BLM 2012 OBJECT ORIENTED PROGRAMMING Sec.1 (2018/2) D012									
Sıra	No	Adı	Soyadı	Q1	Q2	Q3	Q4	Q5	Final Sınavı
1	08011057	FATİH	OKYAY	6	10	10	6	1	33
2	10011057	ALİ KEMAL CEM	ÖZDEMİR	6	1	1	0	0	8
3	10011094	YİĞİT	YANDAŞ	6	10	4	1	5	26
4	11011017	HAKAN CAN	İPEK	6	10	16	2	9	43
5	11011084	ZÜLFÜ	KORKMAZ	G	G	G	G	G	Girmedi
6	12011015	FATİH	ÇOMAK	6	9	17	10	8	50
7	12011048	SALİH MERT	YİĞİT	G	G	G	G	G	Girmedi
8	12011065	NAZIM BURAK	GENÇ	6	10	17	20	12	65
9	12011079	OĞUZHAN	TİRYAKİ	6	4	16	12	0	38
10	13011047	ENES	ÖZKAYA	6	2	14	4	12	38
11	13011089	FATMA BETÜL	KARAPINAR	6	10	16	10	4	46
12	14011021	BETÜL	AKTAŞ	6	10	19	16	12	63
13	14011052	RAHMİ CEMRE	ÜNAL	7	10	16	28	10	71
14	14011063	EMİNE BETÜL	ŞİNAR	6	7	18	8	11	50
15	14011064	YAĞIZ	AKYÜZ	6	4	20	5	10	45
16	14011704	HATİCE	TEKİN	G	G	G	G	G	Girmedi
17	15011003	YUSUF EMİN	KINATAŞ	G	G	G	G	G	Girmedi
18	15011012	UFUK CAN	AKSOY	6	10	16	37	12	81
19	15011024	MUHAMMET CELAL	USTAÖMER	7	10	20	20	12	69
20	15011030	KÜBRA	KARAKUYU	6	10	14	33	7	70
21	15011032	ABDULLAH	GENÇ	4	10	10	21	0	45
22	15011040	MUHAMMED KAĞAN	ÜLKER	9	9	16	21	0	55
23	15011044	DİLARA	SUVEREN	6	9	12	3	12	42
24	15011050	SERRA	SEMİZ	6	8	8	4	0	26
25	15011051	AŞIR FURKAN	KAYACIK	6	10	14	12	9	51
26	15011053	ORÇUN	ÇELİK	6	10	18	14	9	57
27	15011057	BAHADIR	GÜLTEKİN	6	6	2	7	6	27
28	15011058	HALİT	GÖRMEZ	7	7	18	4	8	44
29	15011063	METEHAN	ÇAYLAK	4	1	7	5	10	27
30	15011064	ALTUĞ NUMAN	YILDIZ	4	10	18	13	11	56
31	15011066	ALPEREN ÖNDER	ÖZKAN	2	2	0	0	0	4
32	15011070	FURKAN	ZEREY	10	7	16	14	15	62
33	15011081	ALPER	SELMAN	6	8	17	11	8	50
34	15011702	ÖZGÜR	KAN	10	10	10	13	8	51
35	15011907	HUSEYN	VALIYEV	6	10	16	30	12	74
36	16011002	AYKUT	AKDENİZ	10	10	16	40	12	88
37	16011003	AHMET	AYDIN	6	10	12	5	12	45
38	16011004	SERDAR	YILDIZ	6	10	18	20	12	66
39	16011005	MEHMET SERDAR	ORMANCI	G	G	G	G	G	Girmedi
40	16011006	MUSTAFA	ÖZTÜRK	G	G	G	G	G	Girmedi
41	16011008	CİHAN	KAYA	6	10	20	27	8	71
42	16011010	FURKAN NECATİ	ÜRKMEZ	G	G	G	G	G	Girmedi
43	16011016	AHMET	ELGÜN	6	7	20	10	9	52
44	16011017	MERT	ÖZ	10	10	16	42	12	90
45	16011023	MEHMET HAYRİ	ÇAKIR	10	10	19	28	15	82
46	16011030	İLAYDA	ŞAHİN	3	0	6	0	0	9
47	16011032	FARUK	ARSLAN	7	10	19	30	12	78
48	16011033	YUSUF	ANI	10	10	19	34	15	88
49	16011039	ONUR	AYDEMİR	6	10	19	27	15	77
50	16011043	RECEP FURKAN	KOÇYİĞİT	7	7	20	22	15	71
51	16011048	AHMET	BAĞCI	7	10	15	38	8	78
52	16011057	BÜŞRA	KÜDEN	6	7	20	15	12	60
53	16011061	BERKAY	HAMARAT	6	10	15	19	4	54
54	16011062	EMİN TEYHAN	USLU	6	8	9	19	12	54
55	16011063	BURAK	DURSUNLAR	10	10	20	40	12	92
56	16011066	MEHMET FURKAN	ŞAHİN	7	7	18	36	11	79
57	16011067	KEREM	YOLCU	6	10	20	12	11	59
58	16011068	SELİN	GEZER	G	G	G	G	G	Girmedi
59	16011073	İPEK	KOÇ	6	10	16	16	0	48
60	16011078	EDİZ	AYDOĞAN	G	G	G	G	G	Girmedi
61	16011080	TOYGAR	KAYAŞ	10	10	18	43	12	93
62	16011081	FAZLI	BOZATAY	10	10	18	38	15	91
63	16011082	CEM ÖZGÜR	DEDE	G	G	G	G	G	Girmedi
64	16011085	FATMA	TANRIKULU	6	10	18	12	8	54
65	16011087	CANER	KAYA	10	10	19	33	15	87
66	16011088	İBRAHİM	TIPIRDIK	7	7	20	22	12	68
67	16011090	AHMET	KAYGISIZ	6	4	16	16	12	54
68	16011093	HALİL İBRAHİM	ULUOĞLU	6	7	20	14	8	55
69	16011112	ENES	GÜNDÜZ	2	1	1	0	0	4
70	16011114	SEDAT	AKGÜL	6	9	12	11	4	42
71	16011116	ÖMÜR	ÇİÇEN	G	G	G	G	G	Girmedi
72	16011120	HÜSEYİN MURAT	GEZER	4	10	2	9	0	25
73	16011133	ÜMİT RASİM	TURHAN	6	10	13	11	1	41
74	16011135	UĞURCAN	YILMAZ	G	G	G	G	G	Girmedi
75	16011609	SEYYİD İBRAHİM	GÜLEÇ	6	10	20	40	15	91
76	16011701	ATAKAN	TEKOĞLU	5	8	16	13	0	42
77	16011703	KAAN	AKINTÜRK	G	G	G	G	G	Girmedi
78	16011705	YUNUS EMRE	DEMİR	5	10	16	10	8	49
79	16011906	KHAGANI	KHANKISHIYEV	4	1	5	10	5	25
80	17011001	ELİF	ÖZCAN	6	10	17	22	11	66
81	17011017	MEHMET ALİ	VARGÜN	6	7	17	18	11	59
82	17011024	ELİF	ÖZKAN	6	7	20	20	12	65
83	17011029	SEZİN	BİNER	6	10	0	0	0	16
84	17011045	BEYZA NUR	SEZGİN	6	10	14	10	12	52
85	17011048	TOLGA	ZİFTCİ	10	10	19	37	12	88
86	17011052	HÜSREV	YUMUŞAK	6	10	20	40	9	85
87	17011053	MAHMUT MURAT	AKTAN	6	7	18	32	9	72
88	17011060	YUSUF ETKİN	KIZILDAĞ	6	7	19	28	8	68
89	17011064	OĞUZHAN	SARIALİOĞLU	6	10	14	28	12	70
90	17011066	ALP BİNTUĞ	UZUN	6	5	18	21	12	62
91	17011068	FATMA ZEHRA	ÇETİN	10	10	20	35	12	87
92	17011069	ALPEREN	ÖZER	6	10	20	10	0	46
93	17011072	BERKE KAAN	AÇIKGÖZ	6	10	14	5	9	44
94	17011091	EREN	TERZİOĞLU	7	7	14	28	12	68
95	17011602	SENA	OCAKÇI	6	5	4	0	0	15
96	17011609	İSMAİL	BAYRAM	6	10	20	45	15	96
97	17011611	BETÜL	ÖN	6	10	20	35	12	83
98	17011902	NINI	KVATCHANTIRADZE	7	10	11	38	12	78
99	17011903	NIHAD	GULUZADE	6	7	18	18	0	49
100	17011904	KAMRAN	BALAYEV	5	7	14	19	7	52
101	17011908	MECDEDDİN	HARRAD	6	10	17	40	15	88
102	18011602	CANBERK	GÜLLÜOĞLU	7	10	19	17	10	63
103	18011613	MEHMET EMRE	GÜL	7	10	20	35	12	84
104	E18011005	ABDUFATTOKH	ABDUFATTOKHOV	G	G	G	G	G	Girmedi
105	E18011006	DASTAN	YULDASHEV	G	G	G	G	G	Girmedi
106	E18012004	EMINA	FAJIC	6	1	5	1	5	18
107	E18012005	MERJEMA	MEHMEDOVIC	4	1	5	0	6	16
108	F18011003	HÜSEYİN BERK	BUĞUR	4	6	13	11	13	47
109	13059023	SABİHA YAĞMUR	BAYRAM	6	9	17	28	8	68
110	13067701	TURAN	ÖZEV	G	G	G	G	G	Girmedi
									0
									0

YILDIZ TEKNİK ÜNİVERSİTESİ

ELEKTRİK-ELEKTRONİK FAKÜLTESİ / BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Öğrencinin Adı Soyadı:	**Öğrenci No:**			**İmza:**	
Dersin Adı: BLM2012 Nesneye Yönelik Progr.	**Tarih/Saat:** 27/05/2019 17:00			**Sınav süresi:** 90dk	
Sınav Türü:	**Vize1**	**Vize2**	**Mazeret**	**Final +**	**Bütünleme**
Unvan Ad Soyad (Ders Yürütücüsü):					

QUESTIONS



Answer these questions according to the UML class schema given above. You may need to extract hidden information from the schema and add necessary code while answering the questions.

Question 1 (10 pts): Write the source code of class Capability.

Question 2 (10 pts): Write the source code of class CapabilityException.

Question 3 (20 pts): Write the source code of class Employee. The details of its methods are as follows:

- addCapability: Adds a new capability to the employee. If the employee has already a capability with the same description, this method should generate a CapabilityException.
- changeCapability: Changes the additional information of a capability that the employee already has. If the employee do not already have such a capability, this method should generate a CapabilityException.

Question 4 (45 pts): Write the source code of class Company. The details of its methods are as follows:

- addEmployee: Adds a new employee and returns an object representing the new employee. Any existing employee with the same work ID will be discarded.
- report: This method should execute two tasks simultaneously:
    1. Listing the names and work IDs of its employees to the screen
    2. Saving the employee objects to the disk
    - Hint: You will need additional classes

Question 5 (15 pts): Write the source code of class Main. This class will be used for testing purposes. You are expected to create two employees, add one capability to each employee and print a report by using the namesake method of the company.

BLM2012 OBJECT ORIENTED PROG., Section 1&2, FINAL EXAM ANSWERS (27/05/2019)

Question 1 (10 pts): Write the source code of class Capability.

```
import java.io.Serializable;
public class Capability implements Serializable {
	private static final long serialVersionUID = 1L;
	private String description, additionalInfo;

	public Capability(String description) {
		this.description = description;
	}
	public String getDescription() { return description; }
	public String getAdditionalInfo() { return additionalInfo; }
	public void setAdditionalInfo(String additionalInfo) {
		this.additionalInfo = additionalInfo;
	}
}
```

Question 2 (10 pts): Write the source code of class CapabilityException.

```
public class CapabilityException extends java.io.IOException {
	public CapabilityException( String msg ) {
		super(msg);
	}
}
```

Question 3 (20 pts): Write the source code of class Employee.

```
import java.io.*;
import java.util.*;
public class Employee implements Serializable {
	private static final long serialVersionUID = 1L;
	private String name;
	private Integer workID;
	private LinkedList<Capability> capabilities;

	public Employee( String name, Integer workID ) {
		this.name = name; this.workID = workID;
		capabilities = new LinkedList<Capability>();
	}
	public Integer getWorkID() { return workID; }
	public String getName() { return name; }

	public void addCapability( String desc, String info ) throws CapabilityException {
		Capability newCapability = searchCapability(desc);
		if( newCapability == null ) {
			newCapability = new Capability(desc);
			newCapability.setAdditionalInfo(info);
			capabilities.add( newCapability );
		}
		else
			throw new CapabilityException("Employee #" + workID
				+ " already has capability " + desc + " added.");
	}
	public void changeCapability( String desc, String info ) throws CapabilityException {
		Capability newCapability = searchCapability(desc);
		if( newCapability != null ) {
			newCapability.setAdditionalInfo(info);
		}
		else
			throw new CapabilityException("Employee #" + workID
					+ " does not have capability " + desc + " for changing.");
	}
	public Capability searchCapability( String desc ) {
		for( Capability cap : capabilities )
			if( cap.getDescription().compareToIgnoreCase(desc) == 0 )
				return cap;
		return null;
	}
}
```

Question 4 (45 pts): Write the source code of class Company. Hint: You will need additional classes

```
import java.util.*;
public class Company {
	private Map<Integer,Employee> employees;
	public Company( ) {
		employees = new Hashtable<Integer,Employee>();
	}
	public Employee addEmployee( String name, Integer workID ) {
		Employee newHire = new Employee( name, workID );
		employees.put( workID, newHire );
		return newHire;
	}
	public Employee searchEmployee( Integer workID ) {
		return employees.get( workID );
	}
	public void report( String fileName ) {
		Thread screenThread, fileThread;
		screenThread = new Thread( new ReportToScreen( employees ) );
		fileThread = new Thread( new ReportToFile( fileName, employees ) );
		screenThread.start(); fileThread.start();
	}
}
import java.util.*;
public class ReportToScreen implements Runnable {
	private final Map<Integer,Employee> employees;
	public ReportToScreen(Map<Integer, Employee> employees) { this.employees = employees; }
	public void run() {
		for( Employee anEmployee : employees.values() )
			System.out.println( anEmployee.getWorkID() + " " + anEmployee.getName() );
	}
}
import java.io.*; import java.util.*;
public class ReportToFile implements Runnable {
	private String fileName;
	private final Map<Integer,Employee> employees;
	public ReportToFile(String dosyaAdi, Map<Integer,Employee> employees ) {
		this.fileName = dosyaAdi; this.employees = employees;
	}
	public void run() {
		try {
			ObjectOutputStream output = new ObjectOutputStream(
					new FileOutputStream( fileName ));
			for( Employee calisan : employees.values() )
				output.writeObject( calisan );
			output.close();
		}
		catch( IOException e ) { e.printStackTrace(); }
	}
}
```

Question 5 (15 pts): Write the source code of class Main.

```
public class Main {
	public static void main(String[] args) {
		try {
			Company metaCortex = new Company();
			Employee nihat = metaCortex.addEmployee( "Nihat Genç", 06006 );
			nihat.addCapability("Edebiyat", "Yazarlık düzeyinde.");
			Employee oktay = metaCortex.addEmployee( "Oktay Sinanoğlu", 34034 );
			oktay.addCapability("Kimya", "Profesörlük düzeyinde.");
			metaCortex.report("test.dat");
		}
		catch (CapabilityException e) { e.printStackTrace(); }
	}
}
```

Question 4: Alternative to additional regular classes: Additional inner classes.

```
import java.util.*;
import java.io.*;
public class Company {
      private Map<Integer,Employee> employees;

      public Company( ) {
            employees = new Hashtable<Integer,Employee>();
      }
      public Employee addEmployee( String name, Integer workID ) {
            Employee newHire = new Employee( name, workID );
            employees.put( workID, newHire );
            return newHire;
      }
      public Employee searchEmployee( Integer workID ) {
            return employees.get( workID );
      }
      public void report( String fileName ) {
            Thread screenThread, fileThread;
            screenThread = new Thread( new ReportToScreen( ) );
            fileThread = new Thread( new ReportToFile( fileName ) );
            screenThread.start();
            fileThread.start();
      }
      class ReportToScreen implements Runnable {
            public void run() {
                  for( Employee anEmployee : employees.values() )
                        System.out.println( anEmployee.getWorkID() +
                                    " " + anEmployee.getName() );
            }
      }
      class ReportToFile implements Runnable {
            private final String fileName;

            public ReportToFile(String dosyaAdi) {
                  this.fileName = dosyaAdi;
            }
            public void run() {
                  try {
                        ObjectOutputStream output = new ObjectOutputStream(
                                    new FileOutputStream( fileName ));
                        for( Employee calisan : employees.values() )
                              output.writeObject( calisan );
                        output.close();
                  }
                  catch( IOException e ) {
                        e.printStackTrace();
                  }
            }
      }
}
```