Program 03:

Server.java

```
package serv;

import org.eclipse.californium.core.CoapResource;

import org.eclipse.californium.core.CoapServer;

import org.eclipse.californium.core.server.resources.CoapExchange;

public class server1 extends CoapResource {

        int i=90;

        server1(String str){

                super(str);

}

        @Override

        public void handleGET(CoapExchange exchange) {

                exchange.respond("TEMPT; "+i++);

}

        public static void main(String[]args) {

                CoapServer cs=new  CoapServer();

                cs.add(new server1("TEMPT"));

                cs.start();

}

}
```

Coap.java

```
package serv;


import org.eclipse.californium.core.CoapClient;

import org.eclipse.californium.core.CoapResponse;


public class client {

        public static void main(String[]args) {
```

```
        CoapClient cc=new CoapClient("coap://127.0.0.1/TEMPT");
        String txt;
        try {
            for (int i=0;i<10;i++) {
                CoapResponse cr= cc.get();
                txt=cr.getResponseText();
                System.out.println(txt);
                Thread.sleep(1000);


            }


        } catch (Exception e) {
            e.printStackTrace();
        }
    }


}
```

Program 01

```
package program01;


import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
import java.util.TreeMap;


public class program {
    public static void main(String[]args) {
        Scanner sc=new Scanner(System.in);
        Map<String,String> hmap=new HashMap<String,String>();
        Map<String,String> tmap=new TreeMap<String,String>();
        int ch;
        String State,Capital;
        while(true) {
            System.out.println("1.add 2.displayHashmap 3.displayTreemap ");
            System.out.println("enter ur choice");
            ch=sc.nextInt();
            switch(ch) {
            case 1:
                System.out.println("enter st and cpt");
                State=sc.next();
                Capital=sc.next();
                hmap.put(State, Capital);
                tmap.put(State, Capital);
                break;
            case 2:
                for(Map.Entry<String,String> e:hmap.entrySet()) {

                    System.out.println("State:"+e.getKey()+", Capital:"+e.getValue());

                }break;
            case 3:
                for(Map.Entry<String,String> e:tmap.entrySet()) {
                    System.out.println("State:"+e.getKey()+", Capital:"+e.getValue());

                }break;
            }
        }

    }
}
```

Program02

Create 3 class 1.student

2.employee

3.final

Student.java

```
package program02;

public class Stydent {
	int usn;
	String name;
	public int getUsn() {
		return usn;
	}
	public void setUsn(int usn) {
		this.usn = usn;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	@Override
	public String toString() {
		return "Stydent [usn=" + usn + ", name=" + name + "]";
	}
	public Stydent(int usn, String name) {
		super();
		this.usn = usn;
		this.name = name;
	}


}
```

Employee.java

```
package program02;

public class Employee {
	int id;
	String name;
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	@Override
	public String toString() {
		return "Employee [id=" + id + ", name=" + name + "]";
	}
	public Employee(int id, String name) {
		super();
		this.id = id;
		this.name = name;
	}

}
```

Mani class.java

```
package program02;

import java.util.ArrayList;

import java.util.LinkedList;

import java.util.Scanner;

public class program02 {

	public static void main(String[]args) {

		ArrayList<Stydent> al=new ArrayList<Stydent>();

		LinkedList<Employee> ll=new LinkedList<Employee>();

		Scanner sc=new Scanner(System.in);

		int ch,id;

		String name;

		while(true) {

			System.out.println("1.add student 2.remove student 3.display arraylist 4.add employeeto front 5.add emp to last 6.rem emp from front 7. rem emp from last 8.display ");

			System.out.println("enter ur choice");

			ch=sc.nextInt();

			switch(ch) {

			case 1:

				System.out.println("enter usn and name");

				id=sc.nextInt();
```

```
          name=sc.next();
          al.add(new Stydent(id,name));
          break;
case 2:
          System.out.println("remove student");
          id=sc.nextInt();
          al.remove(id);
          break;
case 3:
          for(Stydent s:al) {
                    System.out.println(s.toString());

          }
          break;
case 4:
          System.out.println("enter id and name");
          id=sc.nextInt();
          name=sc.next();
          ll.addFirst(new Employee(id,name));
          break;
case 5:
          System.out.println("enter id and name");
          id=sc.nextInt();
          name=sc.next();
          ll.addLast(new Employee(id,name));
          break;
case 6:
          ll.removeFirst();
          break;
case 7:
          ll.removeLast();
```

```
                break;
            case 8:
                for(Employee s1:ll) {
                    System.out.println(s1.toString());

                }
                break;

            }

        }
    }

}
```

Hibernate.Java

```
import java.util.List;
import java.util.Scanner;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.boot.Metadata;
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

public class Runner {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        StandardServiceRegistry ssr=new
StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();
        Metadata m=new MetadataSources().buildMetadata(ssr);
        SessionFactory sf=m.buildSessionFactory();
        String name,addr;
        int id,ch;
        student std = new student();
        while(true) {
            Session s=sf.openSession();
            Transaction t=s.beginTransaction();
            System.out.println("1.insert,2.update,3.delete,4.display");
            System.out.println("enter ur choice");
            ch=sc.nextInt();
            switch(ch) {
            case 1:
                System.out.println("enter id ,name and address");
                id=sc.nextInt();
                name=sc.next();
                addr=sc.next();
                std.setId(id);
                std.setName(name);
                std.setAddr(addr);
                s.save(std);
                t.commit();
                break;
            case 2:
                System.out.println("enter id ,name and address");
                id=sc.nextInt();
                name=sc.next();
                addr=sc.next();
                std.setId(id);
                std.setName(name);
                std.setAddr(addr);
                s.saveOrUpdate(std);
                t.commit();
                break;
            case 3:
                System.out.println("enter id");
                id=sc.nextInt();
                std.setId(id);
                s.delete(std);
                t.commit();
                break;
            case 4:
                List<student> l=s.createQuery("from student").list();
                for(student ss:l) {
                    System.out.println(ss.toString());
                }
                break;
            }
        }

    }

}
```

Student.java

```
@Entity
@Table
public class student {
	@Id
	int id;
	String name,addr;
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public String getAddr() {
		return addr;
	}
	public void setAddr(String addr) {
		this.addr = addr;
	}
	@Override
	public String toString() {
		return "student [id=" + id + ", name=" + name + ", addr=" + addr + "]";
	}


}
```