```java
import java.io.Serializable;

public class Address implements Serializable
{
   private String addressLine1;
   private String addressLine2;

   public Address()
   {
      this("");
   }

   public Address(String newAddress)
   {
      String[] addressLines = newAddress.split(",");
      if (addressLines.length == 2)
      {
         setAddressLine1(addressLines[0]);
         setAddressLine2(addressLines[1]);
      }
      else
      {
         setAddressLine1("");
         setAddressLine2("");
      }
   }

   public void setAddressLine1(String aLine1)
   {
      addressLine1 = aLine1;
   }

   public void setAddressLine2(String aLine2)
   {
```

```
35        addressLine2 = aLine2;
36     }
37
38     public String getAddressLine1()
39     {
40        return addressLine1;
41     }
42
43     public String getAddressLine2()
44     {
45        return addressLine2;
46     }
47
48     public String getAddress()
49     {
50        return getAddressLine1() + "\n" + getAddressLine2()+ "\n";
51     }
52
53     public String toString()
54     {
55        return getAddressLine1() + ", " + getAddressLine2();
56     }
57
58     public static void main(String[] args)
59     {
60        Address myAddress = new Address("CS Dept,ThisUni");
61        System.out.println("myAddress = \n" + myAddress);
62
63        Address none = new Address();
64        System.out.println("none = \n" + none.getAddress());
65     }
66  }
```

```java
import java.io.Serializable;

public class Name implements Serializable
{
   private String firstName;
   private String initials;
   private String surname;
   private boolean noInitial;

   public Name()
   {
      this("");
   }

   public Name(String newName)
   {
      String[] nameParts = newName.split(",");
      if (nameParts.length==3)
      {
         setFirstName(nameParts[0]);
         setInitials(nameParts[1]);
         setSurname(nameParts[2]);
         noInitial=false;
      }
      else if(nameParts.length==2)
      {
         setFirstName(nameParts[0]);
         setInitials("");
         setSurname(nameParts[1]);
         noInitial=true;
      }
      else
      {
         setFirstName("");
```

```
35          setInitials("");
36          setSurname("");
37          noInitial=false;
38       }
39
40    }
41    public void setFirstName(String firstName)
42    {
43       this.firstName = firstName;
44    }
45
46    public void setInitials(String initials)
47    {
48       this.initials = initials;
49    }
50
51    public void setSurname(String surname)
52    {
53       this.surname = surname;
54    }
55
56    public String getFirstName()
57    {
58       return firstName;
59    }
60
61    public String getInitials()
62    {
63       return initials;
64    }
65
66    public String getSurname()
67    {
68       return surname;
```

```
69      }
70
71      public String getFullName()
72      {
73          if (noInitial)
74              return getFirstName() + " " + getSurname();
75          else
76              return getFirstName() + " " + getInitials() + " " + getSurname();
77      }
78
79      public String toString()
80      {
81          return getFullName();
82      }
83
84
85      public static void main(String[] args)
86      {
87          Name billy = new Name("William,WH,Bonney");
88          System.out.println("billy = " + billy.getFullName());
89
90          Name jd = new Name("John,Doe");
91          System.out.println("jd = " + jd);
92
93          Name none = new Name();
94          System.out.println("none = " + none.getFullName());
95      }
96  }
```

```java
import java.io.Serializable;

public class Person implements Comparable<Person>, Serializable
{

  private String id;
  private Name name;
  private String email;
  private Address address;

  public Person()
  {
    this("00000000","","",new Address());
  }

  public Person(String id, String name, String email, Address address)
  {
    setId(id);
    setName(name);
    setEmail(email);
    setAddress(address);
  }

  public void setId(String id)
  {
    this.id = id;
  }

  public void setName(String newName)
  {
    name = new Name(newName);
  }

  public void setEmail(String email)
```

```
35  {
36    this.email = email;
37  }
38
39  public void setAddress(Address newAddress)
40  {
41    address = newAddress;
42  }
43
44  public String getId()
45  {
46    return id;
47  }
48
49  public Name getName()
50  {
51    return name;
52  }
53
54  public String getEmail()
55  {
56    return email;
57  }
58
59  public Address getAddress()
60  {
61    return address;
62  }
63
64  public int compareTo(Person other)
65  {
66    String thisField = getName().getSurname()+getId();
67    String otherField = other.getName().getSurname()+other.getId();
68    return thisField.compareTo(otherField);
```

```
69    }
70
71    public String toString()
72    {
73      return "ID: "+getId()+", Name: "+getName().getFullName()+", eMail:
  "+getEmail()+", Address: "+getAddress();
74    }
75
76    public static void main(String[] args)
77    {
78      Address a1 = new Address("My Res,University Town");
79      Person s1 = new
  Person("53456785","Clever,C,Clive","clever.clive@gmail.com",a1);
80      System.out.println("s1: "+s1);
81      System.out.println("Name of s1 = "+s1.getName());
82      System.out.println("and address\n"+s1.getAddress().getAddress());
83
84      Address a2 = new Address("CS Dept,University Place");
85      Person pp = new Person("53476585","Prof,,Piet","prof.piet@gmail.com",a2);
86      System.out.println("pp: "+pp);
87      System.out.println("Name of pp = "+pp.getName().getFullName());
88      System.out.println("and address\n"+a2.getAddress());
89
90      System.out.println("After deleting s1:");
91      s1 = null;
92      System.out.println("Address a1:\n"+a1.getAddress());
93      System.out.println("Name of s1 = "+s1);
94
95    }
96  }
97
```

```java
public class Student extends Person
{

  private String status;

  public Student()
  {
    this("00000000","","",new Address(","));
  }

  public Student(String id, String name, String email, Address address)
  {
    super(id, name, email, address);
  }

 public void setStatus(String status)
  {
    this.status = status;
  }

  public String getStatus()
  {
    return status;
  }

  public String toString()
  {
    return "Student " + super.toString()+", Status: "+getStatus();
  }

  public static void main(String[] args)
  {
    Student s1 = new Student();
    System.out.println("s1: "+s1);
```

```
35
36      Student s2 = new Student("12345678","Clever
    Clive","clever.clive@gmail.com",new Address("My res,My Unitown"));
37      System.out.println("s2: "+s2);
38      s2.setStatus("Senior");
39      System.out.println("s2: "+s2);
40    }
41 }
```

```java
public class Employee extends Person
{

  private String office;

  public Employee()
  {
    this("00000000","","",new Address(","));
  }

  public Employee(String id, String name, String email, Address address)
  {
    super(id, name, email, address);
  }

  public void setOffice(String office)
  {
    this.office = office;
  }

  public String getOffice()
  {
    return office;
  }

  public String toString()
  {
    return super.toString() + ", Office: "+getOffice();
  }

  public static void main(String[] args)
  {
    Employee f1 = new Employee();
    System.out.println("f1: "+f1);
```

```
35
36      Employee f2 = new Employee("12345678","Clever
    Clive","clever.clive@gmail.com",new Address("Borgo Pass 95, Exeter"));
37      System.out.println("f2: "+f2);
38      f2.setOffice("G3, G16");
39      System.out.println("f2 after setting office: "+f2);
40    }
41 }
42
```

```java
public class Faculty extends Employee
{

  private String rank;

  public Faculty()
  {
    this("00000000","","",new Address(","),"");
  }

  public Faculty(String id, String name, String email, Address address, String rank)
  {
    super(id, name, email, address);
    setRank(rank);
  }

  public void setRank(String rank)
  {
    this.rank = rank;
  }

  public String getRank()
  {
    return rank;
  }

  public String toString()
  {
    return "Faculty "+ super.toString()+", Rank: "+getRank();
  }

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

```
34      Faculty f1 = new Faculty();
35      System.out.println("f1: "+f1);
36
37      Faculty f2 = new Faculty("12345678","Clever
    Clive","clever.clive@gmail.com",new Address("My home, My Unitown"),"Lecturer");
38      System.out.println("f2: "+f2);
39      f2.setOffice("G3 G16");
40      System.out.println("f2 after setting office: "+f2);
41
42    }
43  }
44
```

```
1  public class Staff extends Employee
2  {
3
4    private String title;
5
6    public Staff()
7    {
8      this("00000000","","",new Address(","),"");
9    }
10
11   public Staff(String id, String name, String email, Address address, String
     title)
12   {
13     super(id, name, email,address);
14     setTitle(title);
15   }
16
17   public void setTitle(String title)
18   {
19     this.title = title;
20   }
21
22   public String getTitle()
23   {
24     return title;
25   }
26
27   public String toString()
28   {
29     return "Staff "+super.toString()+", Title: "+getTitle()+", Office:
     "+getOffice();
30   }
31
32   public static void main(String[] args)
```

```
33   {
34     Staff f1 = new Staff();
35     System.out.println("f1: "+f1);
36
37     Staff f2 = new Staff("12345678","X,Plez","plez@gmail.com",new
   Address("Borgo Pass 95, Exeter"),"Mrs");
38     System.out.println("f2: "+f2);
39     f2.setOffice("G3 G16");
40     System.out.println("f2 after setting office: "+f2);
41   }
42 }
```

```java
import java.util.Arrays;

public class UniversityPopulation
{
   private Person[] container;
   private int capacity;
   private int numberOfPeople;

   public UniversityPopulation()
   {
      this(5);
   }

   public UniversityPopulation(int capacity)
   {
      container = new Person[capacity];
      setCapacity(capacity);
      setNumberOfPeople(0);
   }

   private void setCapacity(int cap)
   {
      capacity = cap;
   }

   private void setNumberOfPeople(int num)
   {
      numberOfPeople = num;
   }

   public int getCapacity()
   {
      return capacity;
   }
```

```
35
36    public int getNumberOfPeople()
37    {
38       return numberOfPeople;
39    }
40
41    public boolean isFull()
42    {
43       return numberOfPeople == capacity;
44    }
45
46    public void addPerson(Person newPerson)
47    {
48       if (isFull())
49          grow();
50       container[numberOfPeople] = newPerson;
51       setNumberOfPeople(numberOfPeople+1);
52    }
53
54    public Person getPerson(int index)
55    {
56       return container[index];
57    }
58
59    public void sort()
60    {
61       trim();
62       Arrays.sort(container);
63    }
64
65    public void trim()
66    {
67       Person[] temp = new Person[getNumberOfPeople()];
68       for (int i =0; i<getNumberOfPeople(); i++)
```

```
69          temp[i] = container[i];
70       container = temp;
71       setCapacity(getNumberOfPeople());
72    }
73
74    public void grow()
75    {
76       Person[] temp = new Person[getNumberOfPeople()*2];
77       for (int i =0; i<getNumberOfPeople(); i++)
78          temp[i] = container[i];
79       container = temp;
80       setCapacity(getNumberOfPeople()*2);
81    }
82
83
84    public static void main(String[] args)
85    {
86       UniversityPopulation uniList = new UniversityPopulation(10);
87
88       Address a1 = new Address("My Res,University Town");
89       uniList.addPerson(new
Person("53456785","Clever,C,Clive","clever.clive@gmail.com",a1));
90
91       Address a2 = new Address("CS Dept,University Place");
92       uniList.addPerson(new
Person("53476585","Prof,,Able","prof.able@gmail.com",a2));
93
94       System.out.println("Capacity of uniList = " + uniList.getCapacity());
95       System.out.println("Number of people in uniList = " +
uniList.getNumberOfPeople());
96
97       for (int i = 0; i < uniList.getNumberOfPeople(); i++)
98          System.out.println(uniList.getPerson(i));
99
```

```
100         uniList.sort();
101         for (int i = 0; i < uniList.getNumberOfPeople(); i++)
102            System.out.println(uniList.getPerson(i));
103
104
105     }
106 }
```

```java
import java.util.Scanner;

public class PeopleApplication
{
   public static void main(String args[])
   {
      UniversityPopulation uniList = new UniversityPopulation(5);
      Scanner input = new Scanner(System.in);
      String id, name, email, aLine;
      while (input.hasNext())
      {
         id = input.next();
         name = input.next();
         email = input.next();
         aLine = input.nextLine();
         uniList.addPerson(new Person(id, name, email, new Address(aLine)));
      }
      for (int i = 0; i < uniList.getNumberOfPeople(); i++)
         System.out.println(uniList.getPerson(i));

      System.out.print("\nBefore sorting:");
      System.out.println("Capacity of container = " + uniList.getCapacity() + "
and number of items in container = " + uniList.getNumberOfPeople());
      uniList.sort();
      System.out.print("\nAnd after sorting:");
      System.out.println("Capacity of container = " + uniList.getCapacity() + "
and number of items in container = " + uniList.getNumberOfPeople() + "\n");

      for (int i = 0; i < uniList.getNumberOfPeople(); i++)
         System.out.println(uniList.getPerson(i));
   }
}
```

```
s#12345679#Tom,Sawyer#tds@ourmail.com#Whitewash Street 123,St Petersburg#Senior
f#42345677#Prof,V,Able#pva@mail.com#High Street 23,Unitown#Professor#G3 G21
p#42776562#James,F,Sawyer#jfs@mail.com#Fords Way 29,Jasper#Mr#A1 321
p#35865679#Jonathan,Harker#jh@mail.com#Borgo Pass 95, Exeter#Mr#G3 334
s#33876576#Mina,M,Harker#mmh@mail.com#Witby Road 46, Exeter#First year
f#24676578#Quincy,Harker#qh@mail.com#Unires 54, Unitown#Lecturer#B5 123
f#26456575#Katherine,K,Harker#kkh@mail.com#Riley Street 79, North City#Professor#M1 345
s#36545678#Dorian,Gray#dg@mail.com#Basil Way 47, Hallward#Post graduate
s#33455678#Violet,Gray#vg@mail.com#Peanut Place 50, Toontown#Senior
f#23455675#Clay,Quartermain#cq@mail.com#Hughes Street 5, Jonesville#Lecturer#G2 245
```

```java
import java.util.Scanner;
import java.io.*;

public class PeopleApplication
{

   private static Scanner input;
   private static ObjectOutputStream output;

   public static void openInputFile()
   {
      try
      {
         input = new Scanner(new File("People.txt"));
         input.useDelimiter("[#\\n\\r]");
      }
      catch(IOException ioE)
      {
         System.out.println("Cannot open input file");
         System.exit(1);
      }
   }

   public static void openOutputFile()
   {
      try
      {
         output = new ObjectOutputStream(new FileOutputStream("People.ser"));
      }
      catch(IOException ioE)
      {
         System.out.println("Cannot open output file");
         System.exit(1);
      }
```

```
35    }
36
37    public static void closeInputFile()
38    {
39       input.close();  // Scanner close method does not throw an exception,
   check Scanner documentation
40    }
41
42    public static void closeOutputFile()
43    {
44       try
45       {
46          output.close(); // ObjectOutputStream close method does throw an
   exception, check documentation
47       }
48       catch(IOException ioE)
49       {
50          System.out.println("Cannot close output file");
51          System.exit(1);
52       }
53    }
54
55    public static void application()
56    {
57       UniversityPopulation uniList = new UniversityPopulation(5);
58       char type;
59       String id, name, email, aLine;
60       String status, title, office, rank;
61       while (input.hasNext())
62       {
63          type = input.next().charAt(0);
64          id = input.next();
65          name = input.next();
66          email = input.next();
```

```
67          aLine = input.next();
68          if (type == 's')  // Student
69          {
70             status = input.next();
71             Student newStudent = new Student(id, name, email, new
   Address(aLine));
72             newStudent.setStatus(status);
73             uniList.addPerson(newStudent);
74          }
75          else if (type == 'p')  // Staff
76          {
77             title = input.next();
78             office = input.next();
79             Staff newStaff = new Staff(id, name, email, new
   Address(aLine),title);
80             newStaff.setOffice(office);
81             uniList.addPerson(newStaff);
82          }
83          else if (type == 'f')  // Faculty
84          {
85             rank = input.next();
86             office = input.next();
87             Faculty newFaculty = new Faculty(id, name, email, new
   Address(aLine),rank);
88             newFaculty.setOffice(office);
89             uniList.addPerson(newFaculty);
90          }
91          else
92          {
93             System.out.println("Wrong type: "+type+" "+id+" "+name+" "+email+"
   "+aLine);
94          }
95       }
96       for (int i = 0; i < uniList.getNumberOfPeople(); i++)
```

```
97          System.out.println(uniList.getPerson(i));
98       try
99       {
100         for (int i = 0; i < uniList.getNumberOfPeople(); i++)
101            output.writeObject(uniList.getPerson(i));
102         System.out.println("People written to file");
103      }
104      catch(IOException ioE)
105      {
106         System.out.println("Cannot write object to output file");
107         System.exit(1);
108      }
109
110      System.out.print("\nBefore sorting:");
111      System.out.println("Capacity of container = " + uniList.getCapacity() +
   " and number of items in container = " + uniList.getNumberOfPeople());
112      uniList.sort();
113      System.out.print("\nAnd after sorting:");
114      System.out.println("Capacity of container = " + uniList.getCapacity() +
   " and number of items in container = " + uniList.getNumberOfPeople() + "\n");
115
116      for (int i = 0; i < uniList.getNumberOfPeople(); i++)
117         System.out.println(uniList.getPerson(i));
118   }
119
120   public static void main(String[] args)
121   {
122      openInputFile();
123      openOutputFile();
124      application();
125      closeInputFile();
126      closeOutputFile();
127   }
128}
```

```java
import java.io.*;

public class DeserializePeople
{

   private static ObjectInputStream input;
   private static UniversityPopulation uniList;

   public static void openInputFile()
   {
      try
      {
         input = new ObjectInputStream(new FileInputStream("People.ser"));
      }
      catch(IOException ioE)
      {
         System.out.println("Cannot open input file");
         System.exit(1);
      }
   }

   public static void closeInputFile()
   {
      try
      {
         input.close();
      }
      catch(IOException ioE)
      {
         System.out.println("Cannot close input file");
         System.exit(1);
      }
   }

```

```
35    public static void readFromFile()
36    {
37       uniList = new UniversityPopulation(5);
38       Person inputPerson;
39       try
40       {
41          while(true)  // repeat until EOFException is thrown
42          {
43             inputPerson = (Person)input.readObject();
44             uniList.addPerson(inputPerson);
45          }
46       }
47       catch (EOFException eofE)
48       {
49          System.out.println("EOF reached");
50       }
51
52       catch (IOException ioE)
53       {
54          System.out.println("Error reading objects from file");
55       }
56
57       catch (ClassNotFoundException cnfE)
58       {
59          System.out.println(cnfE);
60       }
61
62    }
63
64    public static void application()
65    {
66       for (int i = 0; i < uniList.getNumberOfPeople(); i++)
67          System.out.println(uniList.getPerson(i));
68
```

```
69        System.out.println("\nList of faculty and their rank\n");
70
71        for (int i = 0; i < uniList.getNumberOfPeople(); i++)
72           if (uniList.getPerson(i) instanceof Faculty)
73              System.out.println(uniList.getPerson(i).getName()+": "+
((Faculty)uniList.getPerson(i)).getRank());
74
75
76     }
77
78     public static void main(String[] args)
79     {
80        openInputFile();
81        readFromFile();
82        closeInputFile();
83        application();
84     }
85  }
```