Programming Projects

8-13 Experiment with a simple derivation. Write a constructor for a child class that does not explicitly refer to the constructor of the parent. What happened? Why?

Two different results can occur. The first case occurs when the parent class contains an explicit declaration of a parameterless constructor (the default constructor). In this case, when the object of the child class is instantiated using a constructor that does not explicitly refer to the parents constructor, the default constructor of the parent is called implicitly before executing the childs constructor. The second case occurs when a constructor for the parent class exists, but no default constructor is declared. This configuration will cause a compile time error when compiling the child class.

8-14 Design and implement a set of classes that define the employees of a hospital: doctor, nurse, administrator, surgeon, receptionist, janitor, etc. Include methods in each class that are named according to the services provided by that person and that print an appropriate message. Create a main method to instantiate and exercise several of the classes.

```
//********************************************************************
//
//   Hospital.java      Programming Project      Application
//
//   Authors:  Lewis and Loftus
//
//   Classes:  Hospital
//             Doctor
//             Nurse
//             Administrator
//             Surgeon
//             Receptionist
//             Janitor
//
//********************************************************************

//--------------------------------------------------------------------
//
//  Class Hospital
//
//  Methods:
//
//     public static void main (String[] args)
//
//--------------------------------------------------------------------

class Hospital {

   //=================================================================
   //  Program to demonstrate using various classes related to
   //  a set of hospital employees.
```

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

     Doctor doctor = new Doctor();
     Nurse nurse = new Nurse();
     Administrator administrator = new Administrator();
     Surgeon surgeon = new Surgeon();
     Receptionist receptionist = new Receptionist();
     Janitor janitor = new Janitor();

     doctor.diagnose();
     nurse.care();
     administrator.budget();
     surgeon.diagnose();
     surgeon.operate();
     receptionist.type();
     receptionist.call();
     janitor.clean();

   }  // method main

}  // class Hospital

//-------------------------------------------------------------------
//
//  Class Employee
//
//  Methods:
//
//     public void set_salary(int salary)
//
//-------------------------------------------------------------------

class Employee {

   int salary;

   //=============================================================
   //  Sets the salary for each employee.
   //=============================================================
   public void set_salary (int salary) {
      this.salary = salary;
   }  // method set_salary
```

```
}  // class Employee


//-------------------------------------------------------------------
//
//  Class Doctor
//
//  Methods:
//
//     public void diagnose()
//
//-------------------------------------------------------------------

class Doctor extends Employee {

   //============================================================
   //  Doctors provide a diagnoses.
   //============================================================
   public void diagnose() {
      System.out.println ("Doctor is diagnosing.");
   }  // method diagnose

}  // class Doctor

//-------------------------------------------------------------------
//
//  Class Nurse
//
//  Methods:
//
//     public void care()
//
//-------------------------------------------------------------------

class Nurse extends Employee {

   //============================================================
   //  Nurses care for their patients.
   //============================================================
   public void care() {
      System.out.println ("Nurse is caring.");
   }  // method care

}  // class Nurse

//-------------------------------------------------------------------
```

```
//
//  Class Administrator
//
//  Methods:
//
//     public void budget()
//
//-------------------------------------------------------------------

class Administrator extends Employee {

   //============================================================
   //  Administorators prepare a budget
   //============================================================
   public void budget() {
      System.out.println ("Administrator is budgeting.");
   }  // method budget

}  // class Administrator

//-------------------------------------------------------------------
//
//  Class Surgeon
//
//  Methods:
//
//     public void operate()
//
//-------------------------------------------------------------------

class Surgeon extends Doctor {

   //============================================================
   //  Surgeons are doctors who operate.
   //============================================================
   public void operate() {
      System.out.println ("Surgeon is operating.");
   }  // method operate

}  // class Surgeon

//-------------------------------------------------------------------
//
//  Class Receptionist
```

```
//
//  Methods:
//
//     public void type()
//     public void call()
//
//-------------------------------------------------------------------

class Receptionist extends Employee {

   //=============================================================
   //  Receptionists type reports and forms.
   //=============================================================
   public void type() {
      System.out.println ("Receptionist is typing.");
   }  // method type

   // ============================================================
   // Receptionists make phone calls
   // ============================================================
   public void call() {
      System.out.println ("Receptionist is calling.");
   }  // method call

}  // class Receptionist

//-------------------------------------------------------------------
//
//  Class Janitor
//
//  Methods:
//
//     public void clean()
//
//-------------------------------------------------------------------

class Janitor extends Employee {

   //=============================================================
   //  Janitors clean the hospital.
   //=============================================================
   public void clean() {
      System.out.println ("Janitor is cleaning.");
   }  // method clean

}  // class Janitor
```

8-15 Design and implement a set of classes that define various types of reading material: books, novels, magazines, textbooks, etc. Include data values that describe various attributes of the material: number of pages, favorite character, etc. Include methods that are named appropriately for each class that print a message. Create a main method to instantiate and exercise several of the classes.

Not Provided

8-16 Design and implement a class that simulates a VCR remote control. The class should have various methods for interacting with the TV as well as recording and playing tapes. Extend the VCR remote control class to create a different type of remote control. Add a method to the subclass that provides additional or less functionality for the remote control. Create a main method which instantiates several different remote controls and exercises the various class methods.

```
//********************************************************************
//
//   Remote.java      Programming Project      Application
//
//   Authors:  Lewis and Loftus
//
//   Classes:  Remote
//             VCR
//
//********************************************************************

//--------------------------------------------------------------------
//
//  Class Remote
//
//  Methods:
//
//   public static void main(String[] args)
//
//--------------------------------------------------------------------

class Remote {

   //=================================================================
   //  Simulates the operations of a VCR remote control.
   //=================================================================
   public static void main(String[] args) {

      VCR video = new VCR();

      video.set_channel(6);
```

```
      video.set_volume(4);
      video.program(8, 0, 'p');
      video.record();
      video.play();
      video.stop();
      video.print();

      System.out.println();

   }  // method main

}  // class remote

//-------------------------------------------------------------------
//
//  Class VCR
//
//  Constructors:
//
//     public VCR()
//     public VCR (int vol, int chnl)
//
//  Methods:
//
//     public void print()
//     public void set_channel (int chnl)
//     public void set_volume (int vol)
//     public void program (int hour, int min, char am_pm)
//     public void play()
//     public void stop()
//     public void record()
//
//-------------------------------------------------------------------

class VCR {

   final int MAXVOLUME = 10, MINVOLUME = 0;
   private int prog_hour, prog_min;
   protected int channel, volume;
   protected boolean playing;
   private boolean programmed, recording;
   private char prog_time;

   //=============================================================
   //  Sets the VCR data to the specified values.
   //=============================================================
```

```
public VCR (int vol, int chnl) {

   volume = vol;
   channel = chnl;
   prog_hour = 12;
   prog_min = 0;
   prog_time = 'a';
   programmed = false;
   playing = false;
   recording = false;

}  // constructor VCR

//=============================================================
//  Sets the VCR data to default values.
//=============================================================
public VCR() {

   volume = 5;
   channel = 3;
   prog_hour = 12;
   prog_min = 0;
   prog_time = 'a';
   programmed = false;
   playing = false;
   recording = false;

}  // constructor VCR

//=============================================================
//  Displays the current state of the VCR.
//=============================================================
public void print() {

   System.out.println("Volume level: " + volume);
   System.out.println("Channel: " + channel);

   if (playing)
      System.out.println("VCR is playing.");
   if (recording)
      System.out.println("VCR is recording.");

   if (programmed)
      // Print programming info. Running method with the current
```

```
      //  values will not affect data values.
      program(prog_hour, prog_min, prog_time);
   else
      System.out.println("VCR timer not programmed.");

}  // method print

//============================================================
//  Set the channel of the VCR tuner.
//============================================================
public void set_channel(int chnl) {
   channel = chnl;
   System.out.println("TV set to channel " + channel);
}  // method set_channel

//============================================================
//  Set the volume of the VCR.
//============================================================
public void set_volume(int vol) {

   volume = vol;
   if (volume > MAXVOLUME)
      volume = MAXVOLUME;
   else
      if (volume < MINVOLUME)
         volume = MINVOLUME;

   System.out.println("TV set to volume level " + volume);

}  // method set_volume

//============================================================
//  Sets the time that the VCR should start recording on.
//============================================================
public void program(int hour, int min, char am_pm) {

   programmed = true;
   prog_hour = hour;
   prog_min = min;
   prog_time = am_pm;

   System.out.print("VCR programmed to record at " + hour + ":");
   if (min < 10)
      System.out.print("0" + min);
   else
      System.out.print(min);
```

```
      if (am_pm == 'a')
         System.out.println(" AM, on channel " + channel);
      else
         System.out.println(" PM, on channel " + channel);

   }  // method program

   //===============================================================
   //  Sets the VCR into play mode.
   //===============================================================
   public void play() {

      playing = true;
      recording = false;
      System.out.println("Video tape inserted ...");
      System.out.println("VCR now playing tape ...");

   }  // method play

   //===============================================================
   //  Stops the VCR from playing or recording.
   //===============================================================
   public void stop() {

      playing = false;
      recording = false;
      System.out.println("VCR stopped ...");

   }  // method stop

   //===============================================================
   //  Set the VCR into record mode.
   //===============================================================
   public void record() {

      playing = false;
      recording = true;
      System.out.print("VCR now recording program on channel ");
      System.out.println(channel);

   }  // method record

}  // class VCR
```

8-17 Design and implement a set of classes that keep track of various sports statistics. Have each low level class represent a certain sport. Tailor the services of the classes to the sport in question and move common attributes to the higher level classes as appropriate. Create a main method to instantiate and exercise several of the classes.

Not Provided

8-18 Design and implement a set of classes that keep track of various demographic information about a set of people. Let each class focus on a particular aspect of data collection. Create a main method to instantiate and exercise several of the classes.

Not Provided