```java
// Michael Cohen
// Chapter 6: Programming Challenge 1

import java.util.Scanner;

public class Employee
{
    private String name;
    private int idNumber;
    private String department;
    private String position;

    public Employee()
    {
       String name = "";
       int idNumber = 0;
       String department = "";
       String position = "";
    }

    public Employee(String w, int x, String y, String z)
    {
        this.name = w;
        this.idNumber = x;
        this.department = y;
        this.position = z;
    }

    public String getName()
    {
        return name;
    }

    public void setName(String w)
    {
        this.name = w;
    }

    public int getID()
    {
        return idNumber;
    }

    public void setID(int x)
    {
        this.idNumber = x;
    }

    public String getDepartment()
    {
        return department;
    }

    public void setDepartment(String y)
    {
        this.department = y;
    }

    public String getPosition()
    {
        return position;
    }

    public void setPosition(String z)
    {
        this.position = z;
    }

    public static void main(String[] args)
    {

        Employee sMeyers = new Employee("Susan Meyers", 47899, "Accounting", "Vice President");
        Employee mJones = new Employee("Mark Jones", 39119, "IT", "Programmer");
        Employee jRogers = new Employee("Joy Rogers", 81774, "Manufacturing", "Engineer");

        System.out.println("Employee: " + sMeyers.getName() + "\nID Number: " + sMeyers.getID() + "\nDepartment: " + sMeyers.getDepartment() + "\nPosition: " + sMeyers.getPosition() + "\n_____
        System.out.println("Employee: " + mJones.getName() + "\nID Number: " + mJones.getID() + "\nDepartment: " + mJones.getDepartment() + "\nPosition: " + mJones.getPosition() + "\n_____
        System.out.println("Employee: " + jRogers.getName() + "\nID Number: " + jRogers.getID() + "\nDepartment: " + jRogers.getDepartment() + "\nPosition: " + jRogers.getPosition() + "\n_____

    }
}
```

Compile Messages | jGRASP Messages | Run I/O | Interactions

End | Clear | Help

```
 ----jGRASP exec: java Employee
Employee: Susan Meyers
ID Number: 47899
Department: Accounting
Position: Vice President
_____
Employee: Mark Jones
ID Number: 39119
Department: IT
Position: Programmer
_____
Employee: Joy Rogers
ID Number: 81774
Department: Manufacturing
Position: Engineer
_____

 ----jGRASP: operation complete.
```

Line:9 Col:16 Code:32 Top:1

```java
// Michael Cohen
// Chapter 6: Programming Challenge 7

import java.util.Scanner;

public class Circle
{
    private double radius;
    final double pi = 3.14159;

    public Circle()
    {
    double radius = 0.0;

    }

    public Circle(double radius)
    {
        this.radius = radius;
    }

    public double getRadius()
    {
        return this.radius;
    }

    public double area()
    {
        return (pi * radius * radius);
    }

    public double diameter()
    {
        return (radius * 2);
    }

    public double circumference()
    {
        return (2 * pi * radius);
    }

    public static void main(String[] args)
    {

        Scanner keyboard = new Scanner(System.in);
        System.out.println("Enter a radius for a circle:");
        double radius = keyboard.nextDouble();
        Circle circ = new Circle(radius);

        System.out.println("The radius of the circle is " + circ.getRadius());
        System.out.println("The area of the circle is " + circ.area());
        System.out.println("The diameter of the circle is " + circ.diameter());
        System.out.println("The circumference of the circle is " + circ.circumference());

    }
}
```

LetterCount... | RNGGame.j... | RNGGameE... | SlotMachin... | Q1Q2.java | StarPattern... | Q7.java | SumOfEven... | Q4.java | TestAverag... | SpeedOfSo... | Test.java | Circle.java | Employee.ja...

Compile Messages | jGRASP Messages | Run I/O | Interactions

End | Clear | Help

```
Enter a radius for a circle:
1
The radius of the circle is 1.0
The area of the circle is 3.14159
The diameter of the circle is 2.0
The circumference of the circle is 6.28318

 ----jGRASP: operation complete.

 ----jGRASP exec: java Circle
Enter a radius for a circle:
2
The radius of the circle is 2.0
The area of the circle is 12.56636
The diameter of the circle is 4.0
The circumference of the circle is 12.56636

 ----jGRASP: operation complete.
```

```java
// Michael Cohen
// Chapter 6: Programming Challenge 9

import java.util.Scanner;

public class MonthDays
{
    private int month;
    private int year;
    private int days;

    public MonthDays(int month, int year)
    {
        this.month = month;
        this.year = year;
    }

    public int numberOfDays()
    {
        int days = 0;
        if (month == 1)
        {
           days = 31;
        }
        else if (month == 2)
        {
           if (((year%100 == 0) && (year%400 == 0)) || (year%4 == 0))
           {
               days = 29;
           }
           else
           {
               days = 28;
           }
        }
        else if (month == 3)
        {
           days = 31;
        }
        else if (month == 4)
        {
           days = 30;
        }
        else if (month == 5)
        {
           days = 31;
        }
        else if (month == 6)
        {
           days = 30;
        }
        else if (month == 7)
        {
           days = 31;
        }
        else if (month == 8)
        {
           days = 31;
        }
        else if (month == 9)
        {
           days = 30;
        }
        else if (month == 10)
        {
           days = 31;
        }
        else if (month == 11)
        {
           days = 30;
        }
        else if (month == 12)
        {
           days = 31;
        }
        return days;
    }

    public static void main(String[] args)
    {

        Scanner keyboard = new Scanner(System.in);
        System.out.println("Enter a month (1-12):");
        int month = keyboard.nextInt();
        System.out.println("Enter a year:");
        int year = keyboard.nextInt();
        MonthDays date = new MonthDays(month, year);

        System.out.println(date.numberOfDays() + " days");

    }
}
```

ssages | jGRASP Messages | Run I/O | Interactions

```
 ----jGRASP exec: java MonthDays
Enter a month (1-12):
3
Enter a year:
1
31 days

 ----jGRASP: operation complete.

 ----jGRASP exec: java MonthDays
Enter a month (1-12):
2
Enter a year:
2020
29 days

 ----jGRASP: operation complete.

 ----jGRASP exec: java MonthDays
Enter a month (1-12):
2
Enter a year:
2021
28 days

 ----jGRASP: operation complete.
```

```java
// Michael Cohen
// Chapter 6: Programming Challenge 14

public class Patient
{
    private String name;
    private String address;
    private String phone;
    private String ename;
    private String ephone;

    public Patient(String name, String address, String phone, String ename, String ephone)
    {
        this.name = name;
        this.address = address;
        this.phone = phone;
        this.ename = ename;
        this.ephone = ephone;
    }

    public String getName()
    {
        return name;
    }

    public void setName(String name)
    {
        this.name = name;
    }

    public String getAddress()
    {
        return address;
    }

    public void setAddress(String address)
    {
        this.address = address;
    }

    public String getPhone()
    {
        return phone;
    }

    public void setPhone(String phone)
    {
        this.phone = phone;
    }

    public String getEname()
    {
        return ename;
    }

    public void setEname(String ename)
    {
        this.ename = ename;
    }

    public String getEphone()
    {
        return ephone;
    }

    public void setEphone(String ephone)
    {
        this.ephone = ephone;
    }

}
```

```java
// Michael Cohen
// Chapter 6: Programming Challenge 14

public class Procedure
{
    private String name;
    private String date;
    private String practitioner;
    private double cost;

    public Procedure(String name, String date, String practitioner, double cost)
    {
        this.name = name;
        this.date = date;
        this.practitioner = practitioner;
        this.cost = cost;
    }

    public String getName()
    {
        return name;
    }
    public void setName(String name)
    {
        this.name = name;
    }

    public String getDate()
    {
        return date;
    }
    public void setDate(String date)
    {
        this.date = date;
    }

    public String getPractitioner()
    {
        return practitioner;
    }
    public void setPractitioner(String practitioner)
    {
        this.practitioner = practitioner;
    }

    public double getCost()
    {
        return cost;
    }
    public void setCost(double cost)
    {
        this.cost = cost;
    }
}
```

```java
// Michael Cohen
// Chapter 6: Programming Challenge 14

import java.text.DecimalFormat;

public class PatientChargesDriver
{
    public static void main(String[] args)
    {

        Patient pat0 = new Patient("Michael Abraham Cohen", "123 School Street, Philadelphia, PA 19019", "1234567890", "Peter Griffin", "1326540978");
        Procedure pro1 = new Procedure("Physical Exam", "April 28, 2020", "Dr. Irvine", 250.00);
        Procedure pro2 = new Procedure("X-ray", "April 28, 2020", "Dr. Jamison", 500.00);
        Procedure pro3 = new Procedure("Blood test", "April 28, 2020", "Dr. Smith", 200.00);

        System.out.println( "Name: " + pat0.getName() + "\nAddress: " + pat0.getAddress() + "\nPhone: " + pat0.getPhone() + "\nEmergency Contact: " + pat0.getEname() + ", " + pat0.getEphone() + "\n_____");

        DecimalFormat df = new DecimalFormat("#.00");

        System.out.println( "Procedure name: " + pro1.getName() + "\nDate: " + pro1.getDate() + "\nPractitioner: " + pro1.getPractitioner() + "\nCharge: " + df.format( pro1.getCost() ) + "\n_____");
        System.out.println( "Procedure name: " + pro2.getName() + "\nDate: " + pro2.getDate() + "\nPractitioner: " + pro2.getPractitioner() + "\nCharge: " + df.format( pro2.getCost() ) + "\n_____");
        System.out.println( "Procedure name: " + pro3.getName() + "\nDate: " + pro3.getDate() + "\nPractitioner: " + pro3.getPractitioner() + "\nCharge: " + df.format( pro3.getCost() ) + "\n_____");

    }
}

```

Circle.java | Employee.java | PatientCharges.java | Patient.java | Procedure.java | PatientChargesDriver.java

Compile Messages | jGRASP Messages | Run I/O | Interactions

End | Clear | Help

```
 ----jGRASP exec: java PatientChargesDriver
Name: Michael Abraham Cohen
Address: 123 School Street, Philadelphia, PA 19019
Phone: 1234567890
Emergency Contact: Peter Griffin, 1326540978
_____
Procedure name: Physical Exam
Date: April 28, 2020
Practitioner: Dr. Irvine
Charge: 250.00
_____
Procedure name: X-ray
Date: April 28, 2020
Practitioner: Dr. Jamison
Charge: 500.00
_____
Procedure name: Blood test
Date: April 28, 2020
Practitioner: Dr. Smith
Charge: 200.00
_____

 ----jGRASP: operation complete.
```