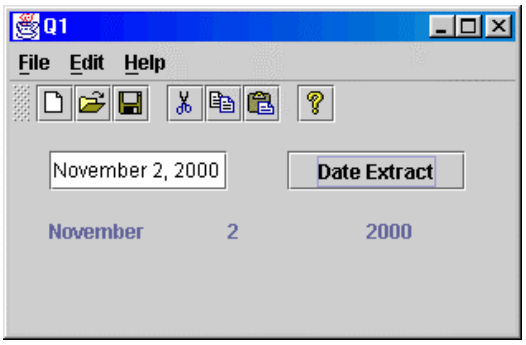
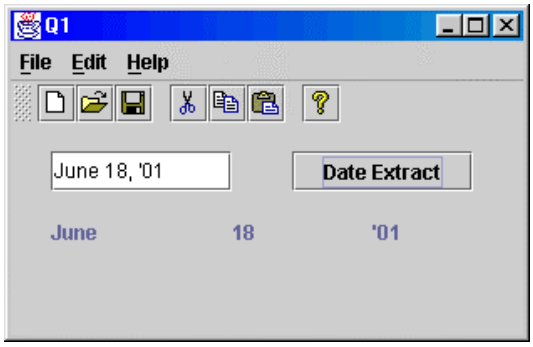
Each question is worth 20 points.

1. You are to write the code that takes a string formatted as

   month day, year

   and extracts the individual components (the month, the day, and the year). None of the three components should have any leading or trailing blanks. For example, consider the following two screen images.





   The following incomplete code segment extracts the month, day, and year from a string stored in a text box named txtDate.

```
void cmdDateExtract_actionPerformed(java.awt.event.ActionEvent event)
{
    String month;
    String day;
    String year;

    String wholeDate = txtDate.getText();

    // Code to perform the extract

    // You supply this code

    // Display results
    lblMonth.setText(month);
    lblDay.setText(day);
    lblYear.setText(year);
}
```

   Fill in the missing code so that the variable month, day, and year are defined as described above. You may assume that the original date is formatted correctly. That is, assume that a space exists between the month and day, and a comma follows the day.

2. Consider the following Java code:

```
void cmdQ2_actionPerformed(java.awt.event.ActionEvent event)
{
    int a = 20;
    int b = 10;
    double c = 2.5;
    double d = 0.5;

    double x;

    x = b / a * c / d;
    System.out.println(String.valueOf(x));   // Part a

    x = c * b % (a - b);
    System.out.println(String.valueOf(x));   // Part b

    x = Math.pow(b , 2 / d);
    System.out.println(String.valueOf(x));   // Part c

    x = (a / b + a / b) * b % a;
    System.out.println(String.valueOf(x));   // Part d
}
```

What value of the variable x is printed for each part (a, b, c, and d)?

3. A palindrome is a word or phrase that reads the same backward or forward. For example "Name no one man".

Assume that the user has entered a phrase into a text field. Write a Java code segment to determine if this phrase is a palindrome. If it is, use a JOptionPane to display the message "It is a palindrome". Otherwise, use a JOptionPane to display the message "It is not a palindrome".

Your code should take into account that fact that the user may enter the phrase with blanks and upper/lower case (just as shown above).

Questions 4 and 5 refer to the Employee/Dependent classes discussed in lecture. The following is a segment of the Employee class:

```
public abstract class Employee
{
   // Instance variables
   private String fEmpNo;
   private String fName;
   private Vector fDependent;
   // Constuctor
   public Employee(String empNo, String name)
   {
      fEmpNo = empNo;
      fName = name;
      fDependent = new Vector();
   }
   // Instance methods
   public String getEmpNo()
   {
      return fEmpNo;
   }
   public void addDependent(Dependent dep)
   {
      fDependent.addElement(dep);
   }
   // more methods exist but are not shown here
```

The definition of the Dependent class is:

```
public class Dependent
{
   // Instance variables
   private String fSSN;
   private String fName;
   private String fSex;
   private int fAge;

   // Constructor
   public Dependent(String ssn, String name, String sex, int age)
   {
      fSSN = ssn;
      fName = name;
      fSex = sex;
      fAge = age;
   }
```

```
    // Accessor methods
    public String getSSN()
    {
       return fSSN;
    }
    public String getName()
    {
       return fName;
    }
    public String getSex()
    {
       return fSex;
    }
    public int getAge()
    {
       return fAge;
    }
}
```

4. Add a method named removeDependent to the Employee class that supports the deletion of an employee's dependent from the vector of dependents. The method should use the dependent's social security number to identify if the dependent is in the vector. The sample code below demonstrates the use of the method.

```
String ssn = txtSSN.getText();
boolean removed = selectedEmp.removeDependent(ssn);
if (removed)
   JOptionPane.showMessageDialog(this, "Found - Dep Removed");
else
   JOptionPane.showMessageDialog(this, "Not Found");
```

5. Modify the Dependent class so that it can store a reference to a single employee. Then add two methods, setEmployee and getEmployee. An example use of the setEmployee method follows:

```
Hourly empH = new Hourly("111", "Sue");
Dependent dep = new Dependent("222-22-2222","Ann","Female",8);
empH.addDependent(dep);
dep.setEmployee(empH);
```

An example use of the getEmployee method is:

```
Employee emp = dep.getEmployee();
```

Exam 1 Sample Answers

1.

```
void cmdDateExtract_actionPerformed(java.awt.event.ActionEvent event)
{
        String month;
        String day;
        String year;

        String wholeDate = txtDate.getText();

        // Code to perform the extract
        int blankPos = wholeDate.indexOf(" ");
        int commaPos = wholeDate.indexOf(",");
        month = wholeDate.substring(0, blankPos).trim();
        day = wholeDate.substring(blankPos+1,commaPos).trim();
        year = wholeDate.substring(commaPos+1, wholeDate.length()).trim();

        // Display results
        lblMonth.setText(month);
        lblDay.setText(day);
        lblYear.setText(year);
}
```

2. a. 0
   b. 5
   c. 10000
   d. 0

3.

```
void cmdPalindrome_actionPerformed(java.awt.event.ActionEvent event)
{
        String s = txtPhrase.getText();
        // shift to lower case
        s = s.toLowerCase();
        // remove all blanks
        String newS = "";
        for (int i=0; i < s.length(); i++)
        {
           if (s.charAt(i) != ' ')
              newS = newS + s.substring(i, i+1);
        }
```

```
            // check for palindrome
            for (int i=0; i < newS.length()/2; i++)
            {
               int k = newS.length() - i - 1;
               if (newS.charAt(i) != newS.charAt(k))
               {
                  JOptionPane.showMessageDialog(this, "Not a palindrome");
                  return;
               }
            }
            JOptionPane.showMessageDialog(this, "It's a palindrome");
      }
```

4.

```
      public boolean removeDependent(String ssn)
      {
            for (int i = 0; i < fDependent.size(); i++)
            {
               Dependent dep = (Dependent)fDependent.elementAt(i);
               if (dep.getSSN().equals(ssn))
               {
                  fDependent.removeElementAt(i);
                  return true;
               }
            }
            return false;
      }
```

5. Add the instance variable:

```
      private Employee fEmp;
```

Add the methods:

```
      public void setEmployee(Employee emp)
      {
       fEmp = emp;
      }

      public Employee getEmployee()
      {
       return fEmp;
      }
```