



EXAMINATIONS — 2003

END-OF-YEAR

COMP 102
INTRODUCTION TO
COMPUTER PROGRAM DESIGN

Time Allowed: 3 Hours

Instructions: Attempt ALL Questions.

Answer in the appropriate boxes if possible — if you write your answer elsewhere, make it clear where your answer can be found.

The exam will be marked out of 180.

Non-programmable calculators without full alphabet keys are permitted.

Non-electronic foreign language dictionaries are permitted.

Questions

	Marks
1. Understanding Java programs	[45]
2. Writing programs with “if”	[15]
3. Arrays	[30]
4. Files	[15]
5. Recursion	[15]
6. Composition	[10]
7. Collections: generic sets	[20]
8. Inheritance	[18]
9. Events and Swing	[12]

Question 1. Understanding Java programs

[45 marks]

(a) [6 marks] What will be printed when the following method is called?

```
public void q1a() {  
    int a = 4;  
    int b = 8;  
    int c = 12;  
    c = b;  
    b = a;  
    a = c;  
    System.out.println( " a is: " + a +  
                        ",b is: " + b +  
                        ",c is: " + c );  
}
```

(b) [6 marks] What will be printed when the following method is called?

```
public void q1b() {  
    int a,b,c;  
    a = 6;  
    b = 9;  
    c = 10;  
    System.out.println( "Calculation 1: " + (a+b*c) );  
    System.out.println( "Calculation 2: " +  
                        (int)(Math.pow(c,2.0) + Math.sqrt( b ) ) );  
    System.out.println( "Calculation 3: " + ((c/b)*a) );  
}
```

(c) [6 marks] Consider the following method definition:

```
public void q1c( int a ) {
    if( a >= 0 ) {
        switch( a ) {
            case 0:
            case 1:
                System.out.println( "Case 1" );
                break;
            case 2:
                System.out.println( "Case 2" );
                break;
            default:
                System.out.println( "Default" );
                break;
        }
    } else {
        if( a <= -5 ) {
            System.out.println( "Blue" );
        } else {
            System.out.println( "White" );
        }
    }
}
```

(i) [3 marks] What will be printed when this method is called with **0** as its argument?

(ii) [3 marks] What will be printed when this method is called with **-1** as its argument?

(d) [6 marks] What will be printed when the following method is called?

```
public void q1d() {  
    int start = 3;  
    int end   = start * 2;  
    for( int c = start; c <= end; c++ ) {  
        System.out.println( c + " " + c * 2 );  
    }  
}
```

(e) [6 marks] What will be printed when the following method is called?

```
public void q1e() {  
    String s = "Hello World\n";  
    String n = "";  
    int    i = 0;  
    while( i < s.length() ) {  
        System.out.print( s.charAt( i ) );  
        i = i + 2;  
    }  
    System.out.println( "\nDone." );  
}
```

(f) [6 marks] What will be printed when the following method is called?

```
public void q1f() {
    String p = "The Quick Brown Fox";
    for( int i = p.indexOf( "Q" ); i < p.length(); i++ ) {
        char c = p.charAt( i );
        if( ! ( c >= 'a' && c <= 'z' ) ) {
            if( c == ' ' ) {
                System.out.print( "+" );
            } else {
                System.out.print( c );
            }
        }
    }
} // end of for loop
}
```

(g) [9 marks] Consider the following program:

```
public void q1g() {
    Person p1 = new Person( "Andrew" );
    Person p2 = new Person( "John", 32 );
    System.out.println( "P1 " + p1.toString() + ", P2 " + p2.toString() );
}
class Person {
    String name = "unknown";
    int age = 21;
    Person( String name, int age ) {
        this.name = name;
        this.age = age;
    }
    Person( String name ) {
        this.name = name;
    }
    public String toString() {
        return name + "(" + age + ")";
    }
}
```

What will this program print out?

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 3. Arrays

[30 marks]

(a) [12 marks] Arrays of Arrays

In this question, there is a class defined with the following private instance variable.

```
private char[][] letters = { { 'a', 'n', 'l', 'r' },  
                             { 'f', 'b', 'a', 's' },  
                             { 'g', 'h', 'c', 'v' },  
                             { 'j', 'z', 'x', '+' } };
```

(i) [4 marks] What will the following method print out?

```
public void q3a1() {  
    System.out.println( letters[3][0] );  
    System.out.println( letters[1][2] );  
    System.out.println( letters[2][3] );  
    System.out.println( letters[0][0] );  
}
```

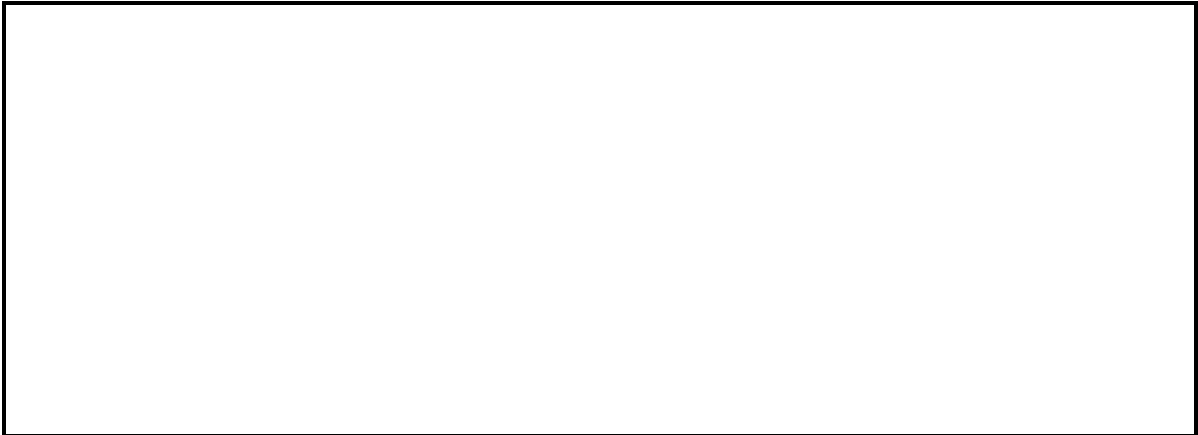
(ii) [4 marks] What will the following method print out?

```
public void q3a2() {  
    char[] a = letters[2];  
    for( int i = a.length-1; i >= 0; i-- ) {  
        System.out.print( a[i] );  
    }  
    System.out.println();  
}
```

(iii) [4 marks]

Write a method which takes a column number as an argument and prints out each of the array elements belonging to that column.

The method should not return any value.



(b) [18 marks] **Array of Objects**

Consider the following program which records and displays the scores from an international sports tournament.

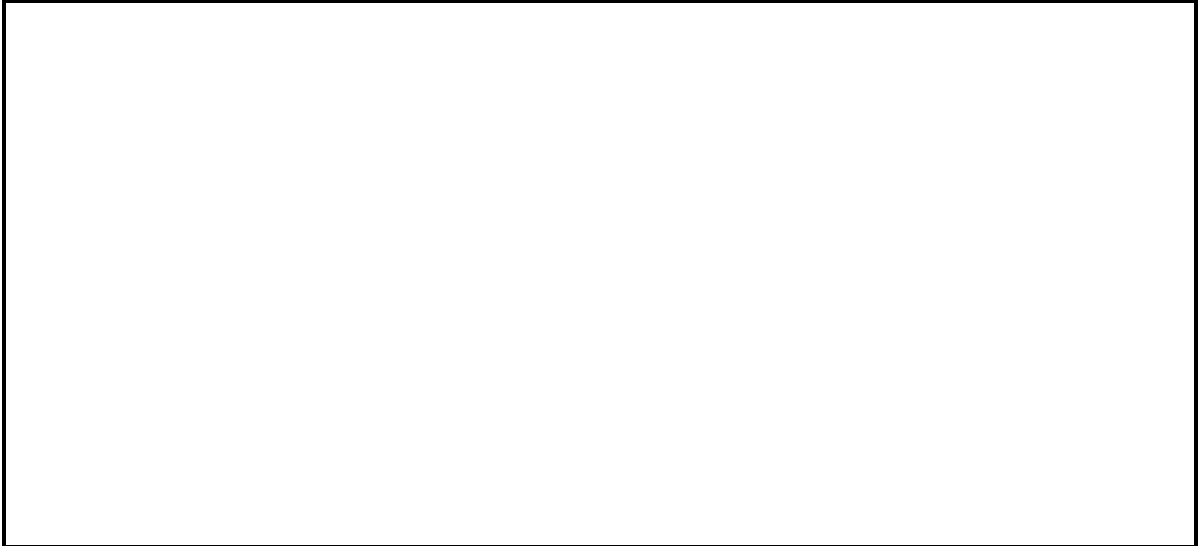
```
public class Question3b {
    public static void main( String[] args ) {
        ResultsTable t = new ResultsTable();
        t.add( "New Zealand", 138, 13 );
        t.add( "Wales"      , 41 , 10 );
        t.add( "Italy"     , 43 , 82 );
        t.add( "Tonga"    , 12 , 36 );
        t.add( "Canada"   , 16 , 109);

        t.display();
    }
}

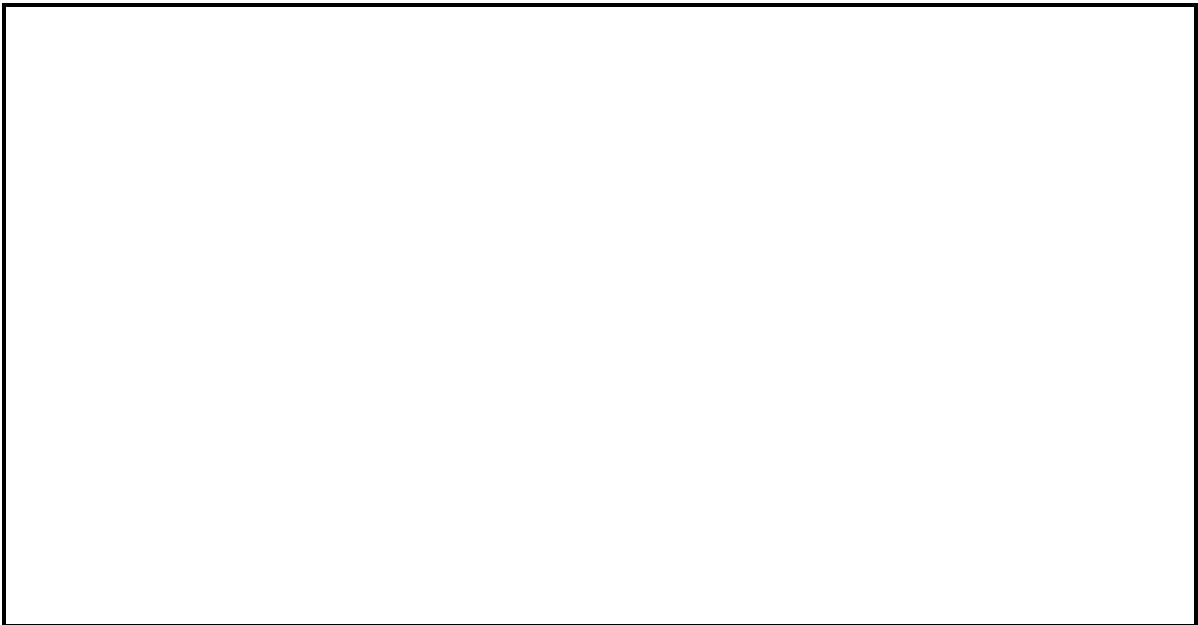
class ResultsRow {
    private String nation;
    private int   pf;      // points for
    private int   pa;      // points against
    public ResultsRow( String n, int pfor, int pagainst ) {
        nation = n;
        pf = pfor;
        pa = pagainst;
    }
    public int getPF() {
        return pf;
    }
    public int getPA() {
        return pa;
    }
    public String getNation() {
        return nation;
    }
    // ..
    // toString method to be completed
    // ..
}

class ResultsTable {
    private ResultsRow[] rows = new ResultsRow[5];
    private int nrows = 0;
    public void add( String n, int pf, int pa ) {
        rows[nrows] = new ResultsRow( n, pf, pa );
        nrows = nrows+1;
    }
    public void display() {
        // ..
        // to be completed
        // ..
    }
}
```

(i) [8 marks] Write a `toString` method for the `ResultsRow` class. The string returned should include the Nation's name, points scored for and points scored against.



(ii) [10 marks] Write a `display` method for the `ResultsTable` class which will print all the scores from the table. Your `display` method **must** make use of the `toString` method you just wrote for the `ResultsRow` class.



Question 4. Reading from Files

[15 marks]

Suppose we have a file containing data about used cars. The data is saved in this format:

year make price

Assume the file `data.txt` contains the following data for six cars.

```
1991 Toyota $5000
2001 Ford $6000
1989 BMW $6000
1997 Toyota $6500
1998 Toyota $8000
1996 Toyota $6000
```

Note: you must answer the following two subquestions (a and b) using the example file given above.

(a) [6 marks] What will the following program print out?

```
import java.io.*;

public class ReadFileEx1 {
    public static void main(String[] args) {
        try {
            FileReader inStream = new FileReader("data.txt");
            BufferedReader ins = new BufferedReader(inStream);
            String dataLine = ins.readLine();
            System.out.println(dataLine);

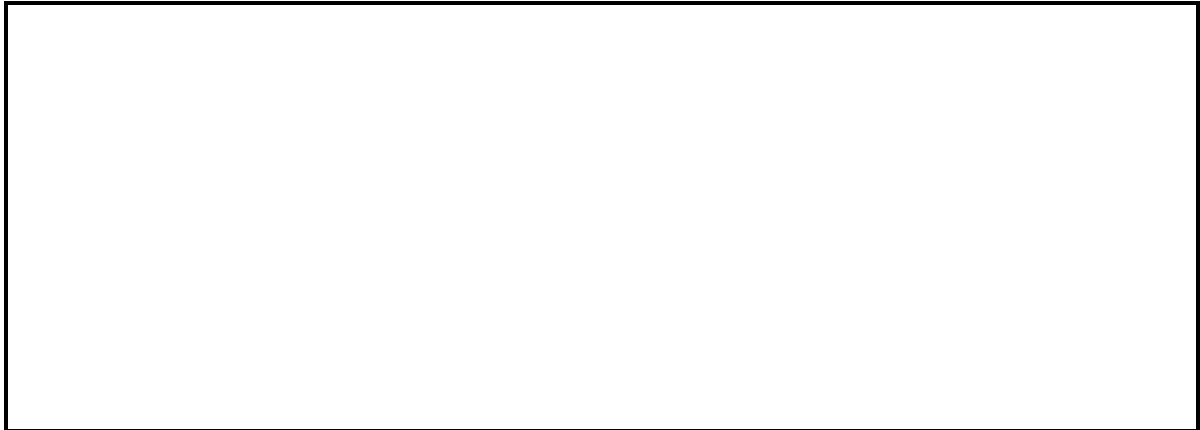
            dataLine = ins.readLine();
            System.out.println(dataLine);

            ins.close();
        }
        catch (IOException ex) {
            System.out.println("i/o error: " + ex.getMessage());
            ex.printStackTrace();
        }
    }
}
```

(b) [9 marks] What will the following program print out?

```
import java.io.*;

public class ReadFileEx2 {
    public static void main(String[] args) {
        try {
            FileReader inStream = new FileReader("data.txt");
            BufferedReader ins = new BufferedReader(inStream);
            String dataLine = ins.readLine();
            int i = 0;
            while (dataLine != null) {
                if (dataLine.indexOf("Toyota") != -1) {
                    String ss = dataLine.substring(dataLine.indexOf("$")+1);
                    double p = Double.parseDouble(ss);
                    if (p < 7000) {
                        i++;
                        System.out.println(i + " " + dataLine);
                    }
                }
                dataLine = ins.readLine();
            }
            ins.close();
        } catch (IOException ex) {
            System.out.println("i/o error: " + ex.getMessage());
            ex.printStackTrace();
        }
    }
}
```



SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 5. Recursion

[15 marks]

Consider the following method:

```
public int myMethod(int n) {  
    System.out.println(n);  
    if (n <= 7)  
        return n;  
    else {  
        int f1 = myMethod(n - 2);  
        int f2 = myMethod(n - 1);  
        int f = f1 + f2;  
        System.out.println(f);  
        return f;  
    }  
}
```

(a) [2 marks] What will this method **return** if it is called with an argument of 8

(b) [13 marks] What will this method **print out** if it is called with an argument of 10? Note that this method always prints the argument and sometimes prints the returned value.

Question 6. Composition

[10 marks]

Consider the following classes:

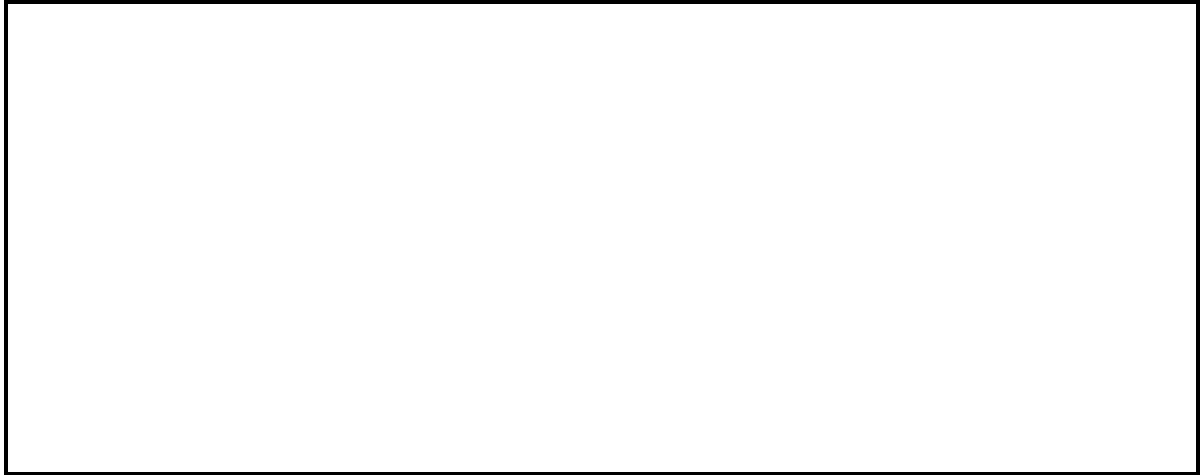
```
class Address {
    private String street, city;
    public Address(String s, String c) {
        street = s; city = c;
    }
    public String makeLabel() { return "To: " + street + ", " + city + ". ";
}
}

class Account {
    private double balance = 0.0, overdraft;
    public Account(double od) { overdraft = od; }
    public boolean deposit(double money) {
        if (money < 0.0) return false;
        balance = balance + money;
        return true;
    }
    public boolean withdraw(double money) {
        if (money < 0.0 || money - balance > overdraft)
            return false;
        balance = balance - money;
        return true;
    }
    public double getBalance() { return balance; }
}

class Customer {
    private String name;
    private Address custadd;
    private Account custacc;
    public Customer(String n, String s, String c) {
        name = n;
        custadd = new Address(s, c);
        custacc = new Account(1000);
    }
    public boolean purchase(double amount) {
        return custacc.withdraw(amount);
    }
    public boolean payment(double amount) {
        return custacc.deposit(amount);
    }
    public String makeBill() {
        String bill;
        bill = custadd.makeLabel();
        bill = bill + "Dear " + name + ", ";
        bill = bill + "Account balance is: " + custacc.getBalance();
        return bill;
    }
}
```

What will the following program print out ?

```
public class Composition {  
    public static void main (String[] args) {  
        Customer c = new Customer("xxx", "yyy", "zzz");  
        System.out.println(c.makeBill());  
        System.out.println(c.payment(500));  
        System.out.println(c.purchase(200));  
        System.out.println(c.makeBill());  
        System.out.println(c.purchase(1500));  
    }  
}
```



Consider the following Generic Set we introduced in the lectures.

```
class Set {
    private Object elements[];
    private int MAX = 5;
    public Set() {
        elements = new Object[MAX];
    }
    public boolean addElement(Object x) {
        int i;
        i = 0;
        while (i < MAX) {
            if (elements[i] == null) {
                elements[i] = x;
                return true;
            }
            i++;
        }
        return false;
    }
    public boolean containsElement(Object x) {
        int i;
        i = 0;
        while (i < MAX) {
            if (elements[i] != null
                && elements[i].equals(x)) {
                return true;
            }
            i++;
        }
        return false;
    }
    public Object findElement(Object x) {
        for (int i = 0; i < MAX; i++) {
            if ((elements[i] != null)
                && (elements[i].equals(x))) {

                return elements[i];
            }
        }
        return null;
    }
    public int size() {
        int count = 0;
        // ..
        // to be completed
        // ..
        return count;
    }
}
```

(a) [6 marks] What will the following method print out?

```
public void test1() {  
    Set s = new Set();  
  
    boolean result;  
    result = s.addElement("Andrew");  
    System.out.println(result);  
  
    System.out.println(s.addElement("Sharon"));  
    System.out.println(s.containsElement("John"));  
  
    String w = (String) s.findElement("Sharon");  
    if (w == null)  
        System.out.println("not found");  
    else  
        System.out.println(w);  
}
```

(b) [8 marks] Add a `size` method to the `Set` class, so it counts how many objects are saved in the collection.

(c) [6 marks] Suppose we have the following class:

```
class Student {
    private String id;
    private String name;
    private String degree;

    public Student( String i, String n, String d ) {
        id = i;
        name = n;
        degree = d;
    }

    public void print() {
        System.out.println("ID: " + id);
        System.out.println(" Name:"+name);
        System.out.println(" Degree:"+degree);
    }

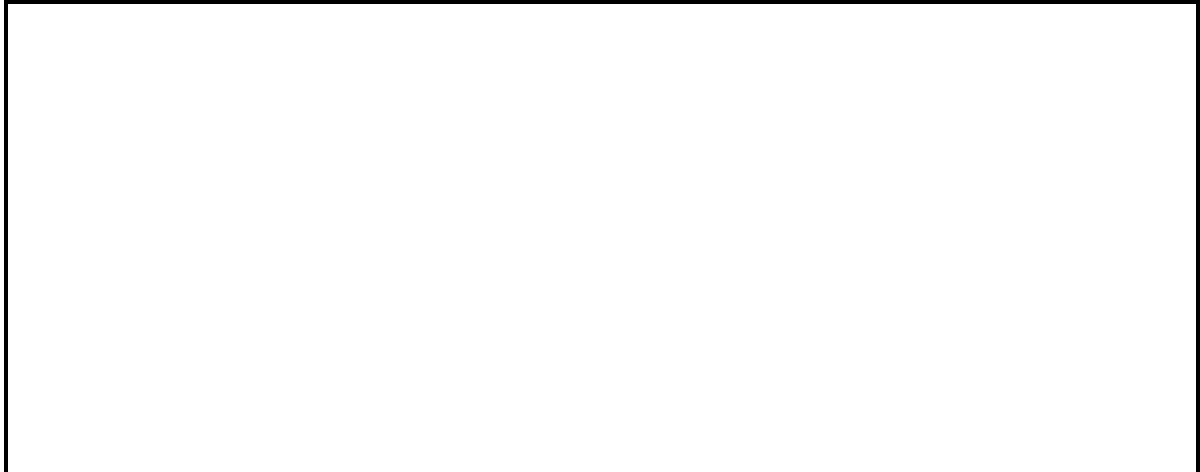
    public String getID() {
        return id;
    }
    public boolean equals(Object x) {
        Student other = (Student) x;
        return id.equals(other.id);
    }
    public String toString(){
        return id+" "+ name +" " + degree;
    }
}
```

What will the following method print out?

```
public void test2() {
    Set s = new Set();
    boolean result;
    Student x, y, z;
    x = new Student("123", "Helen", "BSc");
    result = s.addElement(x);
    System.out.println(result);

    y = (Student) s.findElement(x);
    if (y == null)
        System.out.println("No student!");
    else
        y.print();

    z = (Student) s.findElement(x);
    if (z == null)
        System.out.println("No student!");
    else
        z.print();
}
```



Question 8. Inheritance: Interface and Extension

[18 marks]

Consider the following code:

```
interface Account {
    public void deposit(double money);
    public void withdraw(double money);
    public String getID();
    public double getBalance();
}

class SavingsAccount implements Account{
    private String id;
    private double balance;

    public SavingsAccount(String s) {
        this.id = s;
        this.balance = 0.0;
    }

    public void deposit(double money) {
        this.balance = this.balance + money;
    }

    public void withdraw(double money) {
        this.balance = this.balance - money - 1;
    }

    public String getID() {
        return this.id;
    }

    public double getBalance() {
        return this.balance;
    }
}

class GoalSavingsAccount extends SavingsAccount {
    private int goal;
    public GoalSavingsAccount(String s){
        super(s);
        goal=5000;
    }
    public void withdraw(double money) {
        if (super.getBalance()>= goal){
            super.withdraw(money);
        } else {
            System.out.println("sorry, declined");
        }
    }
}
```

(a) [4 marks] Write down the names of the instance variables (data fields) and methods for the GoalSavingsAccount class, including all instance variables (data fields) and methods inherited.

(b) [6 marks] What will the following program print out?

```
public class BankTesterProgram {  
    public static void main (String[] args) {  
        SavingsAccount s = new SavingsAccount("123");  
        s.deposit(100);  
        s.withdraw(80);  
        System.out.println(s.getBalance());  
        GoalSavingsAccount g = new GoalSavingsAccount("456");  
        g.deposit(2000);  
        g.withdraw(500);  
        System.out.println(g.getBalance());  
        g.deposit(10000);  
        g.withdraw(2000);  
        System.out.println(g.getBalance());  
    }  
}
```


SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 9. Events and Swing

[12 marks]

Consider the following program

```
class TestField implements ActionListener {
    private JFrame    jf;
    private JTextField tf;
    private JButton   jb;
    private JPanel    panel;

    public void actionPerformed((ActionEvent e) ) {
        // part b
    }
    public TestField() {
        jf = new JFrame( "Test Field" );
        tf = new JTextField( 24 );
        jb = new JButton( "Submit" );
        panel = new JPanel();
        panel.add( tf );
        panel.add( jb );
        jf.getContentPane().add( panel );
        tf.addActionListener( this );

        // part a

        jf.setSize(300,100);
        jf.setVisible(true);
    }
}

public class QuestionEvents {
    public static void main( String[] args ) {
        new TestField();
    }
}
```

(a) [4 marks] Events

In the above program, what line of Java needs to be added to make the button respond to click events.

(b) [4 marks]

Assume the `actionPerformed` method will be called when the button is clicked.

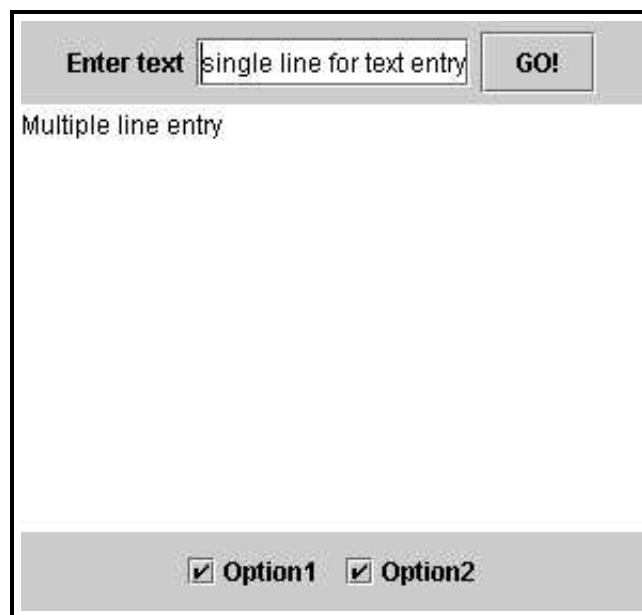
Write a line of Java code for the body of the `actionPerformed` method that will print a message indicating what text was entered in the text field.

eg. If the user had entered "Java is cool !", then clicking the button would cause the following to be printed:

Text Entered was: Java is cool !

(c) [4 marks]

Examine the following diagram.



Write down the names of the Java swing classes that would be needed to create this GUI ?
