Spring 2018 CMP 326 Final Exam V2 ANSWERS

125 points total

Name:______________________ Date:__________

(24 points / 2 points each) **Trace the code and write the result (Write ERROR if there is an ERROR)**

	Code	**Method Call**	**Results**

1	public static int q1(int a, int b) {
	    if (a == 0) {
	        return b;
	    }
	    else {
	        return *q1*(a - 1, b + 1);
	    }
	}
		q1(5,0);	5

		q1(0,3);	3
		q1(3,4);	7
		q1(1,4);	5

2	public static void q2(boolean a, boolean b) {
	    if (a && b) {
	        System.*out*.println("apples");
	    } else if (a || b) {
	        System.*out*.println("bananas");
	    } else {
	        if (!b) {
	            System.*out*.println("bicycles");
	        } else {
	            System.*out*.println("elephants");
	        }
	    }
	}
		q2(true, true);	apples

		q2(false, true);	bananas
		q2(false, false);	bicycles
		q2(true, false);	bananas

3	public static void q3(int [] arr, int i, int x){
	    try{
	        arr[i] = x;
	        System.*out*.println("YES! " + x);
	    }catch(IndexOutOfBoundsException e){
	        System.*out*.println("BOO! ");
	    }catch(Exception e){
	        System.*out*.println("GEN EXC! ");
	    }
	}
		int [ ] a = {7,2,9,10};
		q3(a, 3, 5);
			YES! 5

		int [ ] a = {7,2,9,10};
		q3(a, 2, 11);
			YES! 11

		int [ ] a = {7,9,10,2};
		q3(a, 4, 3);
			BOO!

		int [ ] a = null;
		q3(a, 3, 5);
			GEN EXC!

4 (25 points / 5 points each) **Create the Java class Person that has the following private attributes:**

- **name : String**
- **age : int**
- **gender : char**

a. Create a **default constructor** that initializes all 3 attributes to the following values

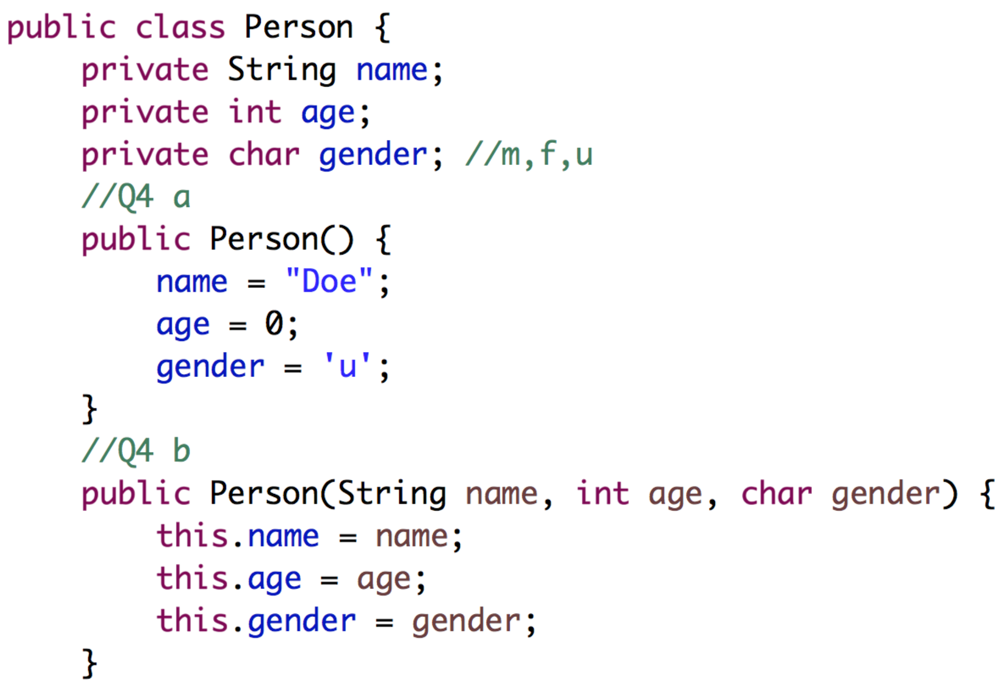
name	“unknown”
age	0
gender	‘u’

b. Create an **overloaded constructor** that takes in values for all 3 attributes and assigns them.

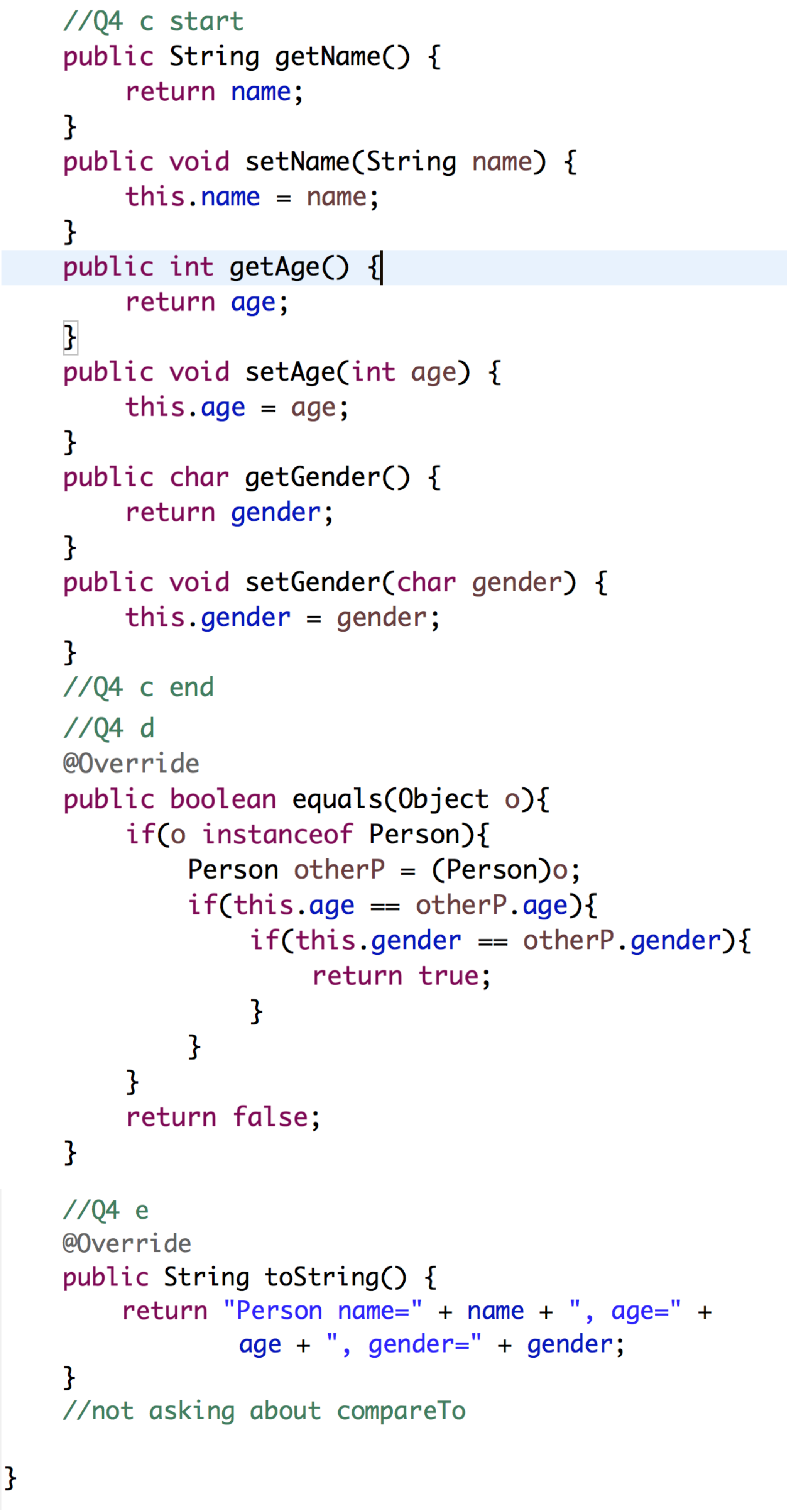
c. Create **getter** and **setter** methods for all 3 attributes.

d. Create the **equals** method.
Two Person objects are equal when their age and gender are the same. (Disregard the names)

e. Create the **toString** method so that it returns a well formatted String including all attributes.



```
public class Person {
    private String name;
    private int age;
    private char gender; //m,f,u
    //Q4 a
    public Person() {
        name = "Doe";
        age = 0;
        gender = 'u';
    }
    //Q4 b
    public Person(String name, int age, char gender) {
        this.name = name;
        this.age = age;
        this.gender = gender;
    }
```



```
    //Q4 c start
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public char getGender() {
        return gender;
    }
    public void setGender(char gender) {
        this.gender = gender;
    }
    //Q4 c end
    //Q4 d
    @Override
    public boolean equals(Object o){
        if(o instanceof Person){
            Person otherP = (Person)o;
            if(this.age == otherP.age){
                if(this.gender == otherP.gender){
                    return true;
                }
            }
        }
        return false;
    }

    //Q4 e
    @Override
    public String toString() {
        return "Person name=" + name + ", age=" +
                age + ", gender=" + gender;
    }
    //not asking about compareTo

}
```

5 (20 points / 5 points each) **Create the Java class Student that inherits from the Person class.**

The Student class has the following private attributes:

- **id : int**
- **gpa : double**
- **major : String**

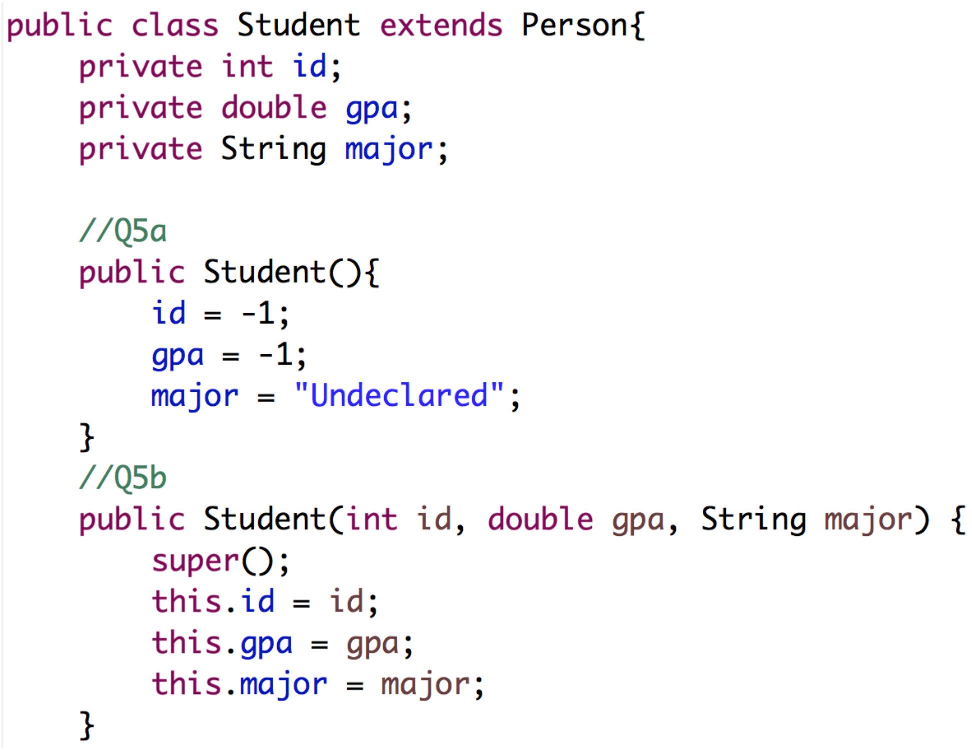
a. Create a **default constructor** that calls the parent constructor and initializes all 3 attributes to the following values

id	-1
gpa	-1.0
major	“undeclared”

b. Create an **overloaded constructor** that takes in values for all 3 attributes and assigns them.

c. Create the **equals** method. Remember to include the super class’s equals method.
Two Student objects are equal when their id is the same. (Disregard the other attributes)

d. Create the **toString** method so that it returns a String including all attributes from both classes.

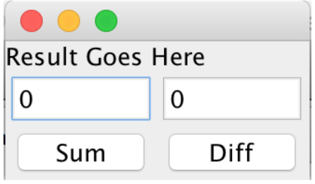


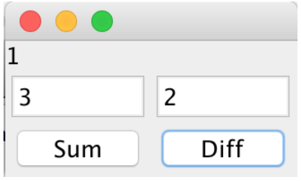
```
public class Student extends Person{
    private int id;
    private double gpa;
    private String major;

    //Q5a
    public Student(){
        id = -1;
        gpa = -1;
        major = "Undeclared";
    }
    //Q5b
    public Student(int id, double gpa, String major) {
        super();
        this.id = id;
        this.gpa = gpa;
        this.major = major;
    }
    //Q5c
    @Override
    public boolean equals(Object o){
        if(o instanceof Student){
            Student otherS = (Student)o;
            if(super.equals(otherS)){
                if(this.id == otherS.id){
                    return true;
                }
            }
        }
        return false;
    }

    //Q5d
    @Override
    public String toString() {
        String s = super.toString();
        s += "\tStudent [id=" + id + ", gpa=" +
                gpa + ", major=" + major + "]";
        return s;
    }
    //not asking for getters and setters
    //not asking about compareTo
}
```

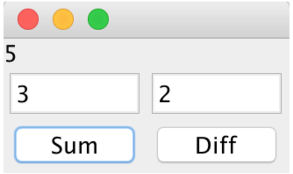
6 (30 points) **Create a GUI for a simple calculator that contains the following elements**:

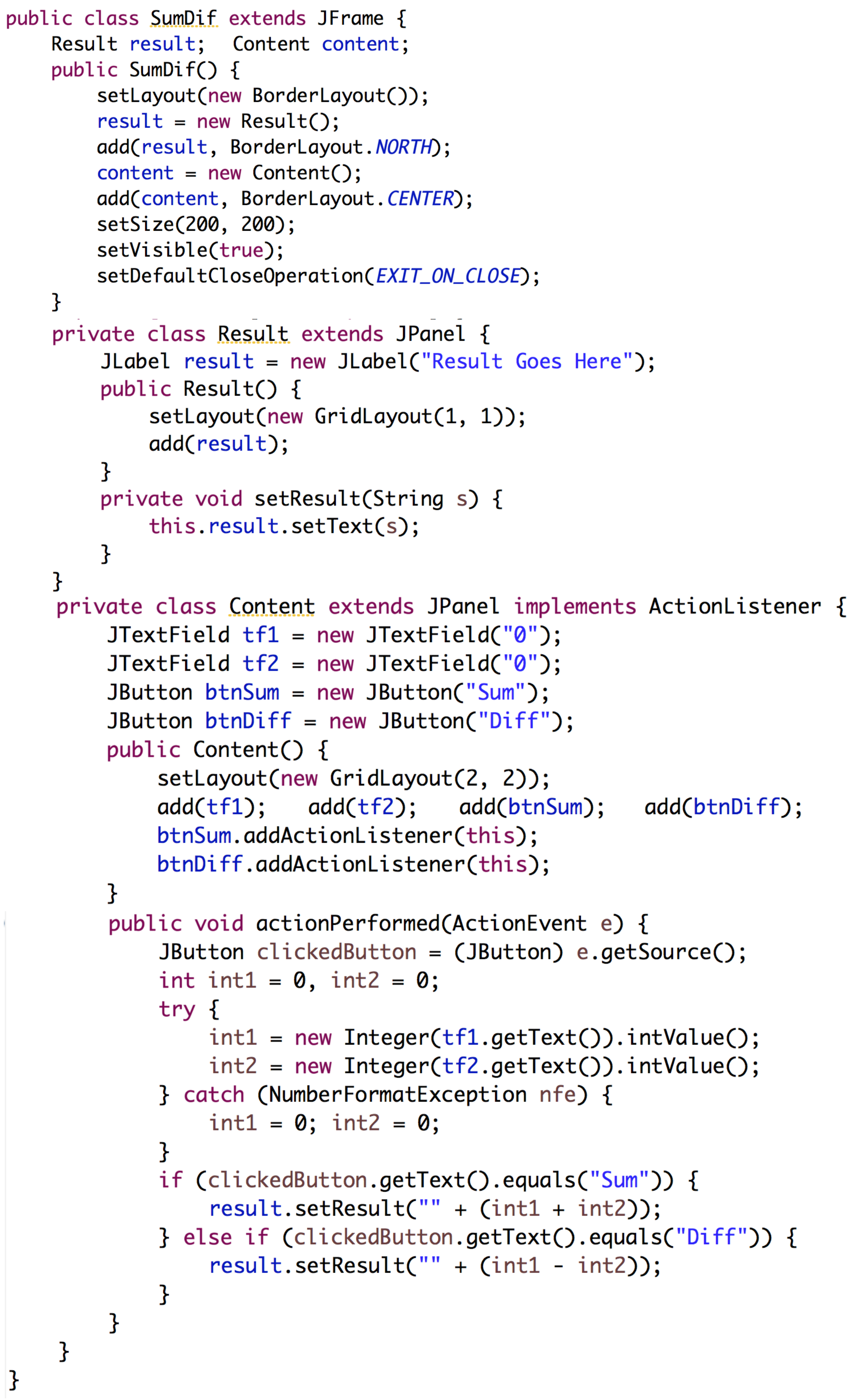
- Two JTextField’s (both start with a 0 as their text)
- Two JButtons (one called “sum” the other called “diff”)
- One JLabel to display the result

The user should be able to use your GUI to enter two numbers using the JTextFields .
If the user presses the Sum button, your GUI should add the two numbers and display the result in the JLabel.
If the user presses the “Diff” button, your GUI should subtract one number from the other and display the result in the JLabel. Remember to catch exceptions (Hint: When a user enters a non-numeric value in a text field it cannot be converted to an int)


Result Goes Here
0 | 0


1
3 | 2

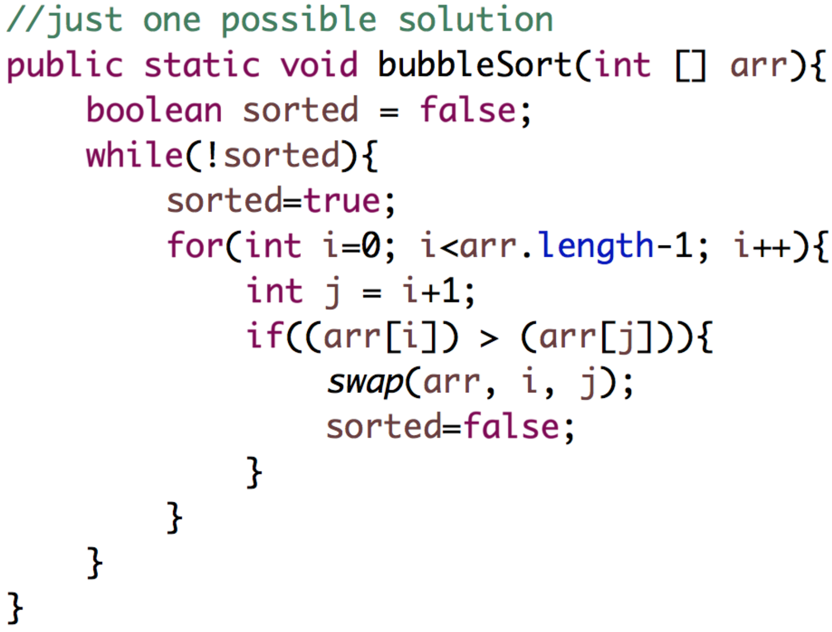

5
3 | 2



```
public class SumDif extends JFrame {
    Result result;  Content content;
    public SumDif() {
        setLayout(new BorderLayout());
        result = new Result();
        add(result, BorderLayout.NORTH);
        content = new Content();
        add(content, BorderLayout.CENTER);
        setSize(200, 200);
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
    private class Result extends JPanel {
        JLabel result = new JLabel("Result Goes Here");
        public Result() {
            setLayout(new GridLayout(1, 1));
            add(result);
        }
        private void setResult(String s) {
            this.result.setText(s);
        }
    }
    private class Content extends JPanel implements ActionListener {
        JTextField tf1 = new JTextField("0");
        JTextField tf2 = new JTextField("0");
        JButton btnSum = new JButton("Sum");
        JButton btnDiff = new JButton("Diff");
        public Content() {
            setLayout(new GridLayout(2, 2));
            add(tf1);   add(tf2);   add(btnSum);   add(btnDiff);
            btnSum.addActionListener(this);
            btnDiff.addActionListener(this);
        }
        public void actionPerformed(ActionEvent e) {
            JButton clickedButton = (JButton) e.getSource();
            int int1 = 0, int2 = 0;
            try {
                int1 = new Integer(tf1.getText()).intValue();
                int2 = new Integer(tf2.getText()).intValue();
            } catch (NumberFormatException nfe) {
                int1 = 0; int2 = 0;
            }
            if (clickedButton.getText().equals("Sum")) {
                result.setResult("" + (int1 + int2));
            } else if (clickedButton.getText().equals("Diff")) {
                result.setResult("" + (int1 - int2));
            }
        }
    }
}
```

7 (10 points) **Remember the iterative Sorting algorithm BubbleSort.**

Create a method that takes in an array of ints and sorts the values in ascending order.



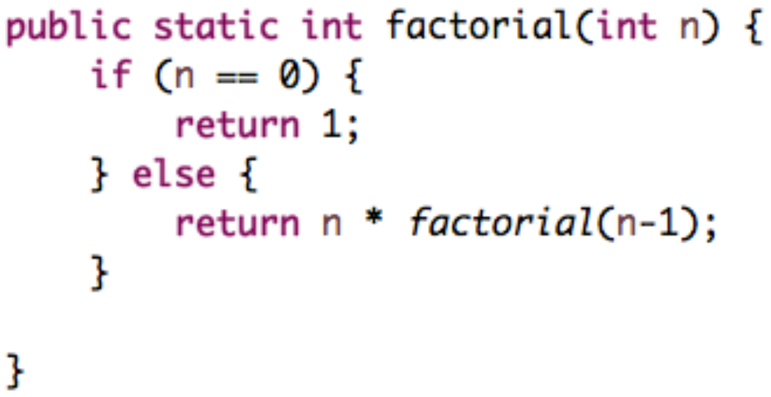
```
//just one possible solution
public static void bubbleSort(int [] arr){
    boolean sorted = false;
    while(!sorted){
        sorted=true;
        for(int i=0; i<arr.length-1; i++){
            int j = i+1;
            if((arr[i]) > (arr[j])){
                swap(arr, i, j);
                sorted=false;
            }
        }
    }
}
```

8 (16 points) **Write a recursive method to calculate factorial of a number n.**

The method should take in an int parameter and return the int factorial of that number.

Example: factorial(5); returns 120

Remember: factorial(0); returns 1



```
public static int factorial(int n) {
    if (n == 0) {
        return 1;
    } else {
        return n * factorial(n-1);
    }

}
```