THE UNIVERSITY OF AUCKLAND

SUMMER SEMESTER, 2009

Campus: City

COMPUTER SCIENCE

Principles of Programming

(Time Allowed: TWO hours)

NOTE: Attempt **ALL** questions
Write your answers in the space provided
There is space at the back for answers that overflow the allotted space
No calculators are permitted

Surname:	
Forenames:	
Student ID number:	
Login name:	

Q1 (/40)	**Q3** (/6)	**Q5** (/9)	**Q7** (/6)
Q2 (/18)	**Q4** (/7)	**Q6** (/6)	**Q8** (/8)
		TOTAL **(/100)**	

ID: .....

Question 1 (40 marks)

a) What output is produced by the following code?

```
System.out.println("1. " + (Math.max(4, Math.min(5, 3))));
```

(2 marks)

b) Which statement in the following program causes a compile error (give the line number)?

```
1   public class Program{
2     public void start(){
3         int x = 3;
4         if(x > 1) {
5             int y = x + 3;
6             x = x + y;
7         }
8         System.out.println(x + ", " + y);
9     }
10  }
```

(2 marks)

ID: .....

c) What output is produced by the following code?

```
String s1 = "I can do this";
String s2 = "I can too";

int pos1, pos2;

pos1 = s1.indexOf("th");
pos2 = s2.indexOf("th");

s2 = s1.substring(pos1);

System.out.println(pos1 + ", " + pos2 + ", " + s2);
```

(2 marks)

d) What output is produced by the following code?

```
Point pt1 = new Point(3, 2);

pt1.x = pt1.y;
pt1.y = pt1.x;

System.out.println(pt1.x + ", " + pt1.y);
```

(2 marks)

ID: .....

e) What output is produced by the following code?

```
Point pt1, pt2;
pt1 = new Point(5, 15);
pt2 = new Point(10, 25);

pt2 = pt1;
pt1 = new Point(pt2);

System.out.println("1.  " + pt1.x + ", " + pt2.x);
System.out.println("2.  " + (pt1 == pt2));
```

(2 marks)

f) What output is produced by the following code?

```
Point pt1, pt2;
pt1 = new Point(5, 15);
pt2 = new Point(7, 15);

pt1.move(2, 0);
System.out.println("1.  " + pt1.equals(pt2));

pt2.translate(2, 0);
System.out.println("2.  " + pt2.x + "," + pt2.y);
```

(2 marks)

ID: .....

g) What output is produced by the following code?

```
Rectangle rect1 = new Rectangle(10, 20, 30, 40);
Rectangle rect2 = new Rectangle(20, 30, 10, 10);

System.out.println("1.  " + rect1.intersects(rect2));

Point pt2 = new Point(70, 10);

System.out.println("2.  " + rect1.contains(pt2));
```

(2 marks)

h) What output is produced by the following program?

```
import java.awt.*;

public class Program {
    public void start() {
        Rectangle r = new Rectangle(10, 20, 30, 40);

        questionH(r);

        System.out.println("1.  " + r.x + ", " + r.y + ", " +
                                    r.width + ", " + r.height);
    }

    private void questionH(Rectangle rect) {
        rect.x = 33;
        rect.width = 15;
    }
}
```

(2 marks)

ID: .....

i) What output is produced by the following program?

```
public class Program {

    public void start() {
        String s = "EXCELLENT!";
        s = method1(s, 6);
        System.out.println("1. " + s);
    }

    private String method1(String s, int x) {
        s = method2(s, x - 1);
        System.out.println("2. " + x);
        return s;
    }

    private String method2(String s, int x) {
        System.out.println("3. " + x);
        return s.substring(0, x);
    }
}
```

(2 marks)

j) Complete the header of the following method.

```
private _____ getInitial (_____) {

    return "" + name.charAt(0);


}
```

(2 marks)

ID: .....

k) Complete the `swapXAndY()` method which swaps the x and y values of the `Point` object passed as a parameter.

```
private void swapXAndY(Point pt) {
    int temp;
}
```

(2 marks)

l) Write a Java boolean expression that tests whether the length of the String variable, `name`, is between 1 and 8 (inclusive).

(2 marks)

m) Write a Java boolean expression that tests whether the String variable, `name`, is equal to "John". (Hint: remember that Strings are objects, not primitive types.)

(2 marks)

ID: .....

n) What output is produced by the following code?

```
int number = 2;

while (number < 5) {
    System.out.println(number * number);
    number++;
}
```

(2 marks)

o) Write a `for` loop that prints out each letter (from left to right) of the String variable, `name`. Each letter should be printed on a separate line.

(2 marks)

ID: .....

p) What is the output of the following program?

```
public class Program {
     public void start() {
         calculateDiscount(36, 500);
     }
     private void calculateDiscount(int age, double amount) {
         if (age > 50 && amount > 200) {
             System.out.println("30% Discount");
         } else if (age > 50) {
             System.out.println("10% Discount");
         } else if (amount > 200) {
             System.out.println("20% Discount");
         } else {
             System.out.println("No Discount");
         }
     }
 }
```

(2 marks)

q) Complete the Java code which prompts the user for their age, reads the user's age (use the `Keyboard` class to do this) and checks if the age entered by the user is greater than or equal to 21. If the age is greater than or equal to 21, the output should be "Age okay", otherwise the output should be "Under age".

```
int age;
System.out.print("Enter age: ");
```

(2 marks)

ID: .....

Question 1 r) – t) are based on the following class.

The `Time24Hr` class definition below is incomplete (incomplete sections are denoted by `...`).

```
public class Time24Hr {

    private int hour;
    private int minutes;
    private int seconds;

    public Time24Hr (int h, int m, int s) {
        hour = h;
        minutes = m;
        seconds = s;
    }

    public ... getMinutes() {
        ...
    }

    public ... setHour(...) {
        ...
    }

    public ... toString() {
        ...
    }
}
```

r) Define the accessor (get) method that returns the minutes of the `Time24Hr` object.

(2 marks)

ID: .....

s) Define the mutator (set) method that updates the hour of the of the `Time24Hr` object.

(2 marks)

t) Define the `toString()` method that returns the time of the `Time24Hr` object in the following format: hour:minutes:seconds. For example, "9:45:16", "10:5:8".

(2 marks)

ID: .....

Question 2 (18 marks)

Consider the following definition of the `Circle` class, which stores the radius and the (x, y) position of a `Circle` object. The (x, y) position represents the centre of the `Circle` object.

```
public class Circle{
    private int radius;
    private int xPosition;
    private int yPosition;
    ...
```

a) Complete the constructor for the `Circle` class.

```
public Circle (int r, int x, int y) {




}
```

(3 marks)

b) Define the `getDiameter()` method which returns the diameter of the `Circle` object. (Note: the diameter of a circle is equal to 2 times the radius.)

```
public int getDiameter() {




}
```

(3 marks)

c) Define the `isBiggerThan()` method which returns `true` if the `Circle` object has a bigger radius than the `Circle` object passed as a parameter to the method.

```
public boolean isBiggerThan(Circle other) {




}
```

(3 marks)

CONTINUED

ID: .....

d) Define the `hasSameCentrePoint()` method which returns `true` if the `Circle` object has the same centre x and y as the `Circle` object passed as a parameter to the method.

```
public boolean hasSameCentrePoint(Circle other) {




}
```

(3 marks)

The following program uses the Circle class defined above.

e) Using the `Circle` class defined above, complete the `start()` method below which prints one of the following statements:

"c1 is bigger" if c1 is bigger than c2,
"c2 is bigger" if c2 is bigger than c1,
"c1 has the same size as c2" if c1 and c2 have the same size.

```
public class Program {

        private Circle getRandomCircle() {
           int r = (int) (Math.random() * 20);
           int x = (int) (Math.random() * 100);
           int y = (int) (Math.random() * 100);

           return new Circle(r, x, y);
        }
        public void start() {

           Circle c1 = getRandomCircle();
           Circle c2 = getRandomCircle();
```

ID: .....

```
        }
    }
```

(6 marks)

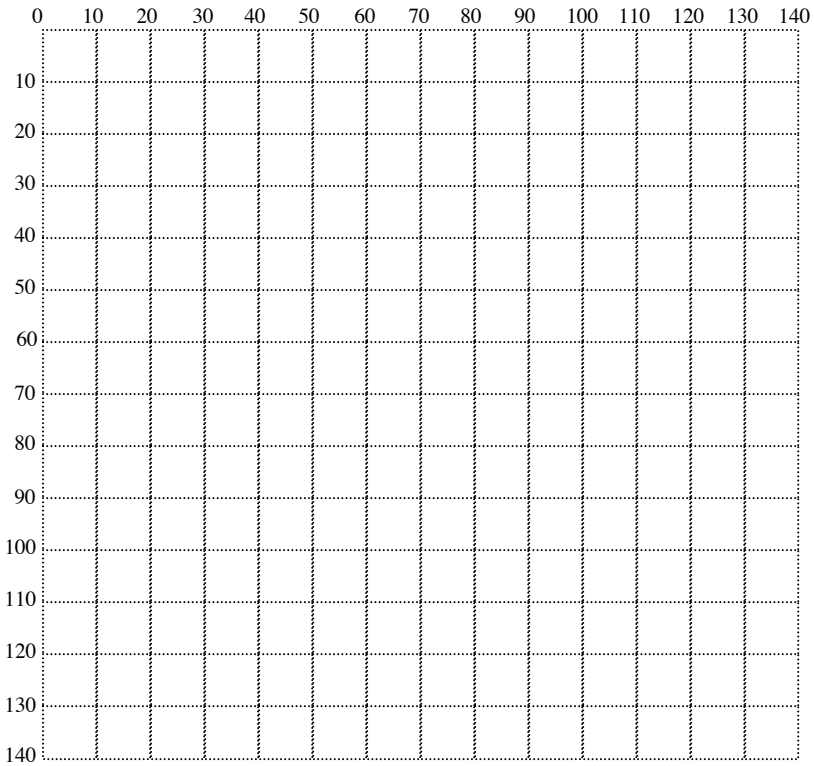
ID: .....

Question 3 (6 marks)

Draw the output produced by the following code in the grid given at the bottom of the page.

The grid lines are not part of the output but are there to help you place the drawing in the correct position. The size of each square in the grid is 10 pixels by 10 pixels.

```
import javax.swing.*;
import java.awt.*;

public class A1JPanel extends JPanel {

  public void paintComponent(Graphics g) {
      super.paintComponent(g);
      draw(g, 60, 70, 20);
  }

  private void draw(Graphics g, int x, int y, int size) {
      g.setColor(Color.BLACK);

      g.drawOval(x, y, size, size);

      g.drawRect(x + size, y + size, size, size);

      g.fillRect(x + size * 2, y + size, size, size);

      g.drawLine(x + size / 2, y + size / 2, x + size * 2, y + size);

      g.drawString("Hi", x + size * 3, y + size);
  }
}
```



(6 marks)

CONTINUED

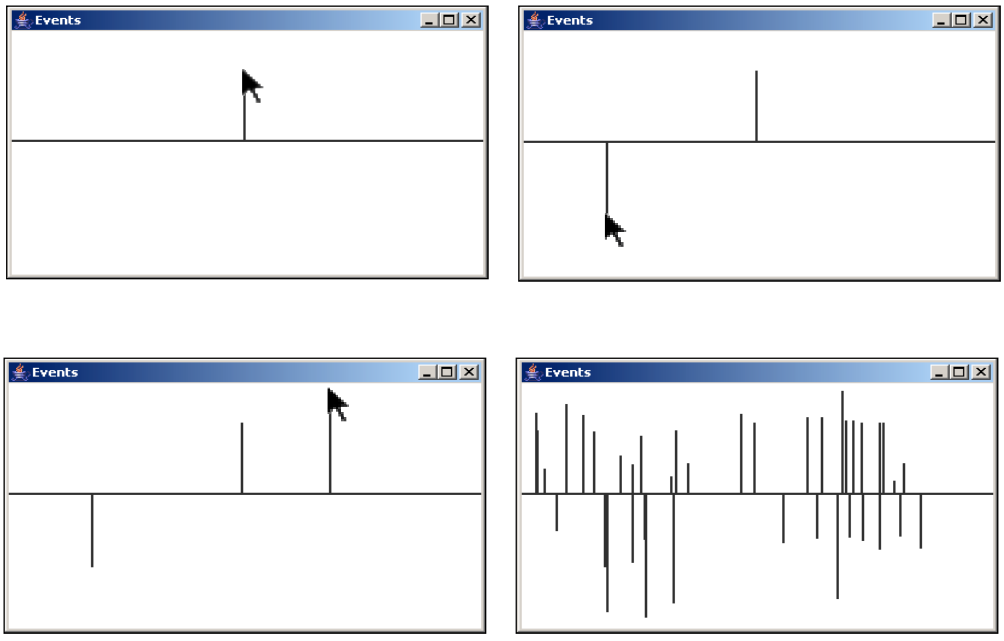
ID: .....

Question 4 (7 marks)

The following `JPanel` responds to MouseEvents. The `JPanel` displays a horizontal line across the `JPanel` from position (0, 100) to position (400, 100). Each time the user presses the mouse a vertical line is drawn from the mouse press position to the horizontal line. The maximum number of vertical lines is given by the constant, `MAX_POINTS`.

Below is a screenshot of the `JPanel` when it is first displayed. Initially there are no vertical lines in the `JPanel`.

Below are some screenshots of the `JPanel` in action. The first, second and third screenshots show the `JPanel` after the user has pressed the mouse in three places and the last screenshot shows the `JPanel` after the user has pressed the mouse many times.



The following numbered statements are part of the code. Place the *correct number* in the *correct place* in the `AJPanel` definition on the next page so that the `AJPanel` executes as described above.

```
1.   lineTops[numberOfPoints] = e.getPoint();
2.   lineTops = new Point[MAX_POINTS];
3.   addMouseListener(this);
4.   private Point[] lineTops;
5.   numberOfPoints++;
6.   repaint();
7.   private static final int MAX_POINTS = 300;
8.   implements MouseListener
```

CONTINUED

ID: .....

```
9.   import java.awt.event.*;
10.  g.drawLine(lineTops[i].x, lineTops[i].y, lineTops[i].x,
                                                    LINE_Y);
11.  public void mouseExited(MouseEvent e) {}
12.  if(numberOfPoints >=  MAX_POINTS) {
       return;
     }
13.  for (int i = 0; i < numberOfPoints; i++) {
14.  }
```

```
import javax.swing.*;
import java.awt.*;


public class AJPanel extends JPanel             {


  private static final int LINE_Y = 100;

  private int numberOfPoints;


  public AJPanel() {
      numberOfPoints = 0;



  }

  public void mousePressed(MouseEvent e) {







  }
  public void paintComponent(Graphics g) {
      super.paintComponent(g);

      g.drawLine(0, LINE_Y, 400, LINE_Y);



  }

  public void mouseClicked(MouseEvent e) {}
  public void mouseEntered(MouseEvent e) {}
  public void mouseReleased(MouseEvent e) {}

}
```

(7 marks)

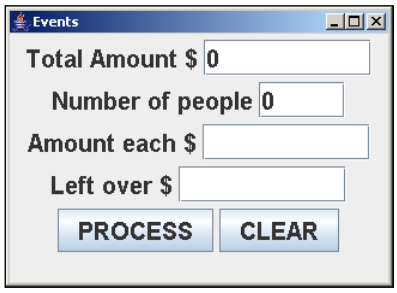
CONTINUED

ID: .....

Question 5 (9 marks)

The `JPanel` defined on the next page contains four `JLabel` components, four `JTextField` components and two `JButton` components.

Below is a screenshot of the `JPanel` when it is first displayed. The `totalAmountT` and the `numberOfPeopleT` JTextFields both display `"0"` and the other two JTextFields are blank.

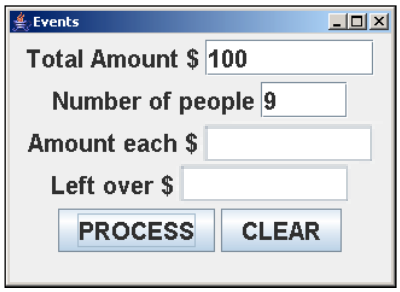


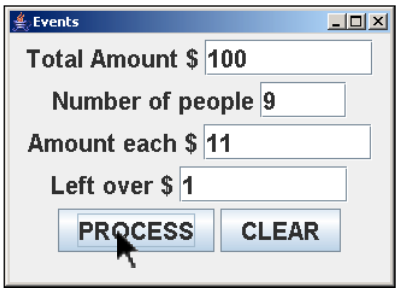
In the `totalAmountT` JTextField, the user enters a whole number, representing the total amount which is to be divided up. In the `numberOfPeopleT` JTextField, the user enters a whole number, representing the number of people who are to receive a portion of the total amount. When the user presses the `'PROCESS'` JButton, the amount each person should receive is displayed in the `amountEachT` JTextField, and the amount left over is displayed in the `leftOverT` JTextField.

When the user presses the `'CLEAR'` JButton, all four JTextFields should display `"0"`.

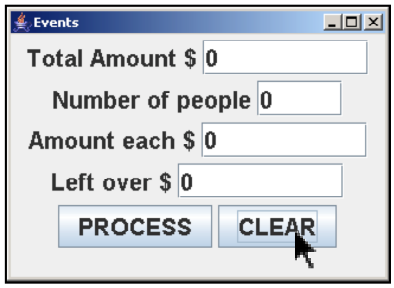
You are required to complete the `JPanel` definition on the next page so that the `JPanel` behaves as described above. You **must** use the variables given in the code.

The screenshot below left shows the `JPanel`, after the user has entered 100 in the `totalAmountT` JTextField and 9 in the `numberOfPeopleT` JTextField. The screenshot below right shows the `JPanel` after the user has pressed the `'PROCESS'` `JButton` (each person should receive $11 and there is $1 left over).





The following screenshot shows the JPanel after the user presses the 'CLEAR' button.



CONTINUED

ID: .....

```
import java.awt.*;
```

```

```

```
import javax.swing.*;

public class AJPanel extends JPanel _____ {

  private JButton processB, clearB;
  private JTextField numberOfPeopleT, totalAmountT, amountEachT,
                                                        leftOverT;
  public AJPanel() {
    JLabel numberOfPeopleL, totalAmountL, amountEachL, leftOverL;
    totalAmountL = new JLabel("Total Amount $");
    numberOfPeopleL = new JLabel("Number of people");
    amountEachL = new JLabel("Amount each $");
    leftOverL = new JLabel("Left over $");

    totalAmountT = new JTextField(8);
    numberOfPeopleT = new JTextField(4);
    amountEachT = new JTextField(8);
    leftOverT = new JTextField(8);

    processB = new JButton("PROCESS");
    clearB = new JButton("CLEAR")
```

```

```

```
    totalAmountT.setText("0");
    numberOfPeopleT.setText("0");

    add(totalAmountL);
    add(totalAmountT);
    add(numberOfPeopleL);
    add(numberOfPeopleT);
    add(amountEachL);
    add(amountEachT);
    add(leftOverL);
    add(leftOverT);

    add(processB);
    add(clearB);
  }
```

```
public void _____ ( _____ e) {
```

CONTINUED

ID: .....

```
}
}
```

(9 marks)

ID: .....

Question 6 (6 marks)

Define the `moveWordsUpArray()` method at the bottom of this page. This method is passed two parameters, an array of Strings and the number of Strings currently in the array. The method moves all the Strings in the array up one position, i.e., the String in position 1 is moved into position 0, the String in position 2 is moved into position 1, and so on.

Consider the code segment below:

```
String[] words = new String[500];
words[0] = "cat";
words[1] = "bat";
words[2] = "rat";
words[3] = "dog";
words[4] = "pig";
words[5] = "ape";
int numberOfWords = 6;

System.out.print("1. List of words: ");
printArray(words, numberOfWords);
System.out.println();

moveWordsUpArray(words, numberOfWords);
numberOfWords--;
System.out.print("2. After removing the first word: ");
printArray(words, numberOfWords);
System.out.println();

moveWordsUpArray(words, numberOfWords);
numberOfWords--;
System.out.print("3. After removing the first word: ");
printArray(words, numberOfWords);
```

If you have defined the `moveWordsUpArray()` method correctly, the output from the above code segment should be:

```
1. List of words: cat bat rat dog pig ape
2. After removing the first word: bat rat dog pig ape
3. After removing the first word: rat dog pig ape
```

Note: The `printArray()` method is not shown here.

```
private void moveWordsUpArray(String[] words, int numberOfWords) {
```

CONTINUED

ID: .....

}

(6 marks)

ID: .....

Question 7 (6 marks)

Complete the `findIndexOfMax()` method which uses a `for` loop to find the **index** of the maximum mark in the array of integers passed as a parameter to the method. For example, if the largest number in the array is the number in position 4 of the array, the method should return 4.

Note: the array passed as a parameter contains at least one element.

```
private int findIndexOfMax(int[] marks) {









}
```

(6 marks)

CONTINUED

ID: .....

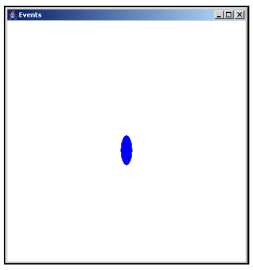
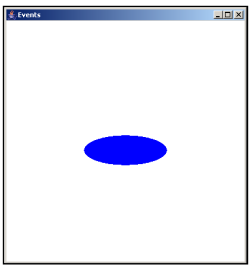
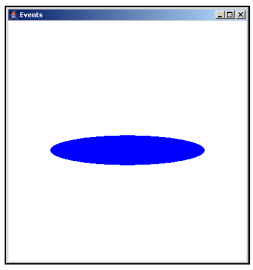
Question 8 (8 marks)

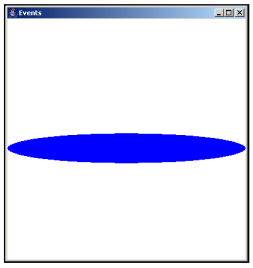
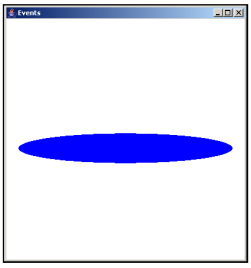
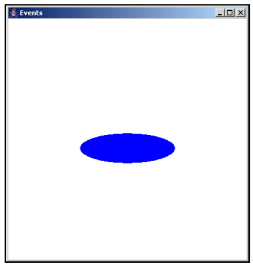
You are required to complete the following `JPanel` which uses a `Timer` object. The `Timer` object is created with a delay of 100 milliseconds. The `JPanel` displays a blue oval with centre at 200, 200, width 20 (given by the constant, `MIN_WIDTH`) and height 50. Once the Timer has started, the oval increases in width across the JPanel until the width of the oval reaches 400 pixels (given by the constant, `MAX_WIDTH`). When the oval has reached the maximum width, the oval should decrease in width until it reaches the initial width of 20 pixels. The amount by which the oval increases/decreases in width each tick of the Timer is 20 pixels.

Whenever the user presses the RIGHT arrow key the `Timer` should start and the oval should continue increasing/decreasing in width across the JPanel. Whenever the user presses the LEFT arrow key the `Timer` should stop and the oval should stop in its current position.

Below are some screenshots of the `JPanel` in action. Screenshot 1 shows the `JPanel` when it is first displayed. Screenshots 2, 3 and 4 show the `JPanel` as the oval increases in width across the JPanel. Screenshots 4 and 5 shows the oval after it has reached the maximum width and has started decreasing in width.

NOTES:
In your code you MUST use the instance variables and constants which have been defined.
You should not define any other instance variables or constants.

1. 
2. 
3.  …

4. 
5. 
6.  …

ID: .....

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class AJPanel extends JPanel  _____

                                     _____  {

  public static final int MAX_WIDTH = 400;
  public static final int MIN_WIDTH = 20;

  private boolean isIncreasing;
  private Rectangle r;
  private Timer t;

  public AJPanel() {
      r = new Rectangle(200 - MIN_WIDTH/2, 175, MIN_WIDTH, 50);
      isIncreasing = true;
```

```
  }

  public void actionPerformed(ActionEvent e) {
      final int CHANGE_AMOUNT = 20;
```

CONTINUED

ID: .....

```
  }
  public void paintComponent(Graphics g) {
         super.paintComponent(g);
```

```
  }
  public void keyPressed(KeyEvent e) {
```

```
  }
  public void keyReleased(KeyEvent e) {}
  public void keyTyped(KeyEvent e) {}
}
```

(8 marks)

CONTINUED

ID: .....

OVERFLOW PAGE

(If you have used this page, please indicate clearly under the relevant question that you have overflowed to this page)

ID: .....

ROUGH WORKING (WILL NOT BE MARKED)

(You may detach this page from the answer booklet and use it for rough working)

ID: .....

ROUGH WORKING (WILL NOT BE MARKED)

(You may detach this page from the answer booklet and use it for rough working)