2001

American University of Beirut

Faculty of Arts & Sciences *Department of Mathematics*

Math 200 – Introduction to Programming

Midterm - Spring-2001

Time: 1h 30 min

Student Id Number	*Student Name*		*Professor's name*
	Last Name	*First Name*	

NO CREDITS WILL BE GIVEN IF THE JUSTIFICATION IS MISSING WHEN ASKED

A. General – 18 points

1. The UltraEdit: (2 pts)

   (a) Compiles programs and stores them in memory.
   (b) Creates programs and stores them in memory
   (c) Creates programs and stores them on disk
   (d) translates the .java programs into .class files.

2. Which of the following are invalid identifier(s)? Justify your answer (3 pts)

   (a) oNe
   (b) my Window
   (c) 123price
   (d) Give2you
   (e) public
   (g) AccountIn$

3. What are the values of the Java variables a and b after the following code is executed: (2 pts)

```
b = false;
a = !b;
if (b || a)
b = !b;
else
a = !b;
```

   (a) a is false and b is false
   (b) a is false and b is true
   (c) a is true and b is false
   (d) a is true and b is true

4. In the space provided, write **T** or **F** to indicate whether the statement is true or false (2 pts)

   - _____ It is an error if the file name is not identical in both spelling and capitalization to the class name with the *.java* file name extension.
   - _____ After the first trial of compilation, even if you have syntax errors, the .class file is generated and stored on the disk.

5. Assuming z has the value 5. The statement
   `System.out.println( " 2 + 3 + z = " + 2 + 3 + z );`
   displays (3 pts)

   (a) 2 + 3 + z = 5 + z
   (b) 2 + 3 + z = 2 + 3 + z
   (c) 2 + 3 + z = 10
   (d) 2 + 3 + z = 235
   (e) 2 + 3 + z = 23z

6. What is the result of the following expression? 4 % 5 + 3 * 4 / (double) 5 + 1
   (3 pts)
   (a) 7.0
   (b) 7.4
   (c) 3.4
   (d) 4.66
   (e) none of the above

7. Which of the following are wrong instance methods? Justify (3 pts)

```
(a) public sqrt()              { return ( 1) ; }
(b) public void int m1()       { return ( 1) ; }
(c) public boolean radius()    { return ( 1==1) ; }
(d) public void m3()           { return; }
```

B. Find the bugs – 12 points

The following application is supposed to read 2 double numbers, compute their average and output the result.
The code as given contains 6 errors.

```
import javax.swing.*;
public class Average {
        public static void main (String[ ] args) {
            double x, y;
            double z;
            String x1 = JOptionPane.showMessageDialog( "enter the first number");
            x= Integer.parseInt(x1);
            double z = x+y /2.0;
            System.exit(0);
            }
    }
```

List the errors indicating for each wether it is a syntax error, a runtime error, or a logical error.

Error1: .....

.....

Kind : .....

Error2: .....

.....

Kind : .....

Error3: .....

.....

Kind : .....

Error4: .....

.....

Kind : .....

Error5: .....

.....

Kind : .....

Error6: .....

.....

Kind : .....

C. Output – 26 points

1. Assuming a = 1, b = 9, and c = 3. What is the output of the following piece of code? (3 pts)

```
if ( a == 1 )
if ( b == 2 )
if ( c == 3 )
System.out.println( "Line1" );
else
System.out.println( "Line2" );
else
System.out.println( "Line3" );
System.out.println( "Line4" );
```

2. Assuming a = 9, b = 9, and c = 3. What is the output of the following piece of code? (3 pts)

```
if ( a == 1 )
System.out.println( "Line1" );
else
if ( b == 2 )
if ( c == 3 )
System.out.println( "Line2" );
else
System.out.println( "Line3" );
System.out.println( "Line4" );
```

3. Assuming a = 6 and b = 8. Modify the following code to produce the output shown. You **may not** make any changes other than inserting braces and changing indentation of the code. (3 pts)

```
if (a == 10) { }
if (b == 7)
System.out.println("@@@@@"); }
else
System.out.println("#####");
System.out.println("$$$$$");
System.out.println("&&&&&");
```

Output

```
#####
$$$$$
&&&&&
```

4. Assume that x=4, and y=7. What are the values of x and y after the following if statement. (3pts)

```
if (x>12 || x++<5 && --y<7)
{
    x++;
    --y;
}
```

.....

.....

.....

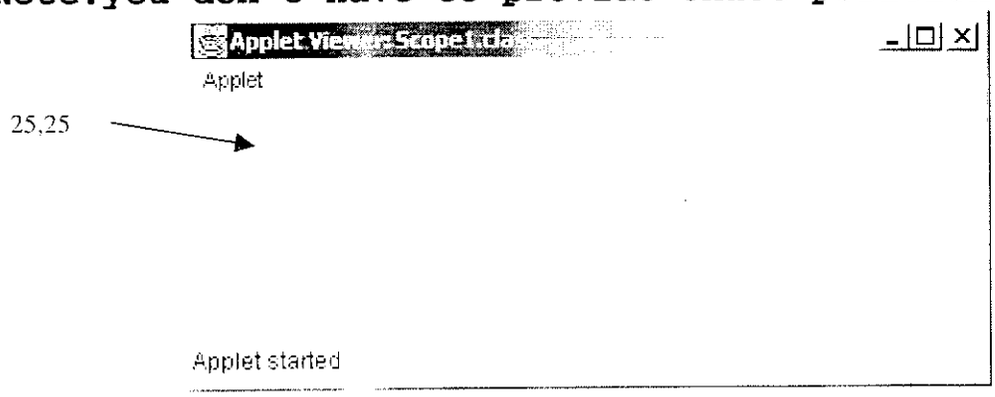
.....

5. What is the output of the applet Scoping? (8 pts)

```
import javax.swing.JApplet;
import java.awt.Graphics;

public class Scope1 extends JApplet {

 int y=25, x=25;

public void paint (Graphics g){

   g.drawString("point at (" + x +","+ y +")",x,y);
   y=y+10;
   method1(g);
   method2(g);
   method1(g);
   g.drawString("point at (" + x +","+ y +")",x,y);
}


public void method1(Graphics g)
 {
   g.drawString("point at (" + x +","+ y +")",x,y);
     g.drawString("point at (" + x +","+ y +")",x+100,y);
     y=y+10;
 }

public void method2(Graphics g)
 {
     int x=10;
        g.drawString("point at (" + x +","+ y +")",x,y);
     x*=10;
     g.drawString("point at (" + x +","+ y +")",x,y);
     y+=10;
 }

}
```

SEE NEXT PAGE

Note:you don't have to provide exact positions!



6. Consider the Java class below: (6 pts)

```
public class TryIt
   private int num1;
   private int num2;

   public TryIt(int param1, int param2) {
     num1 = param1;
     num2 = param2;   }

   public void first(int param1) {
     int a;
     int b;
     a = param1;
     b = a + 2;
     display(a, b);
     second(param1);
     display(this.num1, this.num2); }

   private void second(int param1) {
     int a;
     int b;
     a = param1 + 5;
     b = a + 10;
     this.num2 = this.num2 + 5;
     display(a, b);     }

   private void display(int num1, int num2) {
     System.out.print(num1);
     System.out.print(" ");
     System.out.print(num2);
     System.out.print(" ");
     System.out.print(this.num1);
     System.out.print(" ");
     System.out.println(this.num2);
    }
}
```

What is the output of the following application Test?

```
public class Test {
   public static void main( String args[] ) {
     TryIt var1;
     var1 = new TryIt(5,10);
     var1.first(10);
    }
 }
```

D. Rewrite – 5 points

Rewrite the following *if* statement using a *switch* statement

```
if ((grade > 8) && (grade < 11)) {
a=1; b= 2;
}
else if (grade == 8){
a= 3; b=4;
}
else a= 5;
```

E. Programming – 39 points

1. When you say you are 18 years old, you are really saying that the earth has circled the sun eighteen times. Since other planets take less or more days than the earth to travel around the sun, your age would be different on other planets. You can compute how old you are on other planets by the formula: (15 pts)

$$y = \frac{x \times 365}{d}$$

where x is the age on earth, y is the age on planet Y, and d is the number of earth days the planet Y takes to travel around the sun.

Write an application that inputs the user's earth age, a planet number (1 for Mercury, 2 for Venus, 3 for Jupiter, 4 for Saturn), and prints out his/her age on the chosen planet. The values for **d** are listed in the table below:

Planet	d = approximate number of Earth days for this planet to travel around the sun
Mercury	88
Venus	225
Jupiter	4380
Saturn	10767

```
public class Age {

public static void main (String args [] ) {

//constant declaration (for the 4 values of d)
.....
.....
.....
.....

// variable declarations
.....
.....

// user's input
.....
.....
.....
.....
```

//compute the new age

.....

.....

.....

.....

.....

.....

.....

.....

//output the result

.....

.....

.....

}

}

2. Define a class called **Date** that will keep track of a date represented in the standard fashion as three integers: day (between 1 and 31), month (between 1 and 12), and a four-digit year (e.g., 2000). (14 pts)

Fill in this class definition with the following methods:

```
class Date {

  // instance variables
  private int day, month, year;


   // constructor that takes 3 integer parameters d, m, and y to form a date

   .....

   .....

   .....

   .....



   // instance method setDate
   // It takes 3 parameters d, m, y and update the date(day , month, and year) of an
   // existing object.  it does not return any value


   .....
```

.....

.....

// instance method **getDay**
// It has no parameters and returns the day as an integer

.....

.....

.....

.....

.....

//instance method **getMonth**
//it has no parameters and returns the month as an integer

.....

.....

.....

// instance method **getYear**
// it has no parameters and returns the year as an integer

.....

.....

.....

// instance method **print**
// it has no parameters and prints the date in standard format:(day/month/year)
// using the getDay, getMonth, and getYear methods

.....

.....

.....

.....

.....

// instance method **before**
// it takes one parameter of type Date and returns true if this date's year
// comes before the other

.....

.....

.....

.....