

INSTITUTE OF ACCOUNTANCY ARUSHA



DEPARTMENT OF INFORMATICS

ITT07322: OBJECT-ORIENTED PROGRAMMING

LAB WORKSHEET-01 PROBLEM SOLUTIONS.

PROBLEM SOLUTION 01-19.

<< [Download source-code](#) >>

Table of Contents

Topics: Introduction to OOP and Java Operators.....	3
Problem 01.....	3
Problem 02.....	3
Problem 03.....	3
Problem 04.....	4
Option 01. Using ternary operator.....	4
Option 02. Using if-else.....	5
Problem 05.....	5
Option 01. Using user pi.	5
Option 02. Using built-in standard pi.	6
Topic: Control Statements.....	6
Problem 06.....	6
Problem 07.....	6
Problem 08.....	7
Problem 09.....	7
Problem 10.....	8
Topic: Arrays.....	8
Problem 11.....	8
Problem 12.....	8
Problem 13.....	9
Problem 14.....	9
Problem 15.....	10
Note.....	11
Topic: Classes and objects.	11
Problem 16.....	11
Option 01.In a Single file.	11
Option 02.....	11
Problem 17.....	12
Option 01. In a Single file.	12
Option 02.....	12
Problem 18.....	12
Option 01. In a single file.....	13
Option 02. In a separate files << Download source-code >>	14
Topic: Inheritance.....	14
Problem 19.....	14
Option 01. In a single file.....	15
Option 02. In a separate file << Download source-code >>	15

Topics: Introduction to OOP and Java Operators.

Problem 01. Modify the Java first program so that the program displays your name.

```
MyName.java X
Solutions OOP > problem 01 > MyName.java
1
2
3 /**
4  * @author T.cruiz.
5  * @since Nov `22.
6  */
7 public class MyName{
8     public static void main(String[] args) {
9         System.out.println("MyName SirName");
10    }
11 }
```

Problem 02. Write Java application that displays the sum of two integers.

```
MyName.java SumOfTwo.java X
Solutions OOP > problem 02 > SumOfTwo.java
1 /**
2  * @author T.cruiz.
3  * @since Nov `22.
4  */
5 public class SumOfTwo{
6     public static void main(String[] args) {
7         int a, b, sum;
8         a = 11;
9         b = 8;
10        sum = a + b;
11        System.out.println("The sum of two integer is " + sum);
12    }
13 }
```

Problem 03. Write Java application that obtains two integers typed by a user at the keyboard, computes the sum of these values and outputs the result.

```
MyName.java SumOfTwo.java SumOfInput.java X
Solutions OOP > problem 03 > SumOfInput.java

5
6 import java.util.Scanner;
7
8 public class SumOfInput{
9     public static void main(String[] args) {
10         int a, b, sum;
11         Scanner put = new Scanner(System.in);
12         System.out.println("Enter the numbers :");
13         a = put.nextInt();
14         b = put.nextInt();
15         sum = a + b;
16         System.out.println("The sum of two integer is " + sum);
17     }
18 }
```

Problem 04. Write an application that inputs three integers from the keyboard and prints the sum, average, product and the biggest number of these numbers.

Option 01. Using ternary operator.

```
Arithmetic.java X
Solutions OOP > problem 04 > Arithmetic.java

5
6 import java.util.Scanner;
7
8 public class Arithmetic{
9     public static void main(String[] args) {
10         int a, b, c, sum, prod, avg, bigger;
11         Scanner put = new Scanner(System.in);
12         System.out.println("Enter the numbers :");
13         a = put.nextInt();
14         b = put.nextInt();
15         c = put.nextInt();
16         sum = a + b + c;
17         avg = sum / 3;
18         prod = a*b*c;
19
20         bigger = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c); // using ternary operator
21
22         System.out.println("The sum of three integer is " + sum);
23         System.out.println("The average of three integer is " + avg);
24         System.out.println("The product of three integer is " + prod);
25         System.out.println("The bigger of three integer is " + bigger);
26     }
27 }
```

Option 02. Using if-else.

```
Arthmetical.java X
Solutions OOP > problem 04 > Arthmetical.java
5 import java.util.Scanner;
6 public class Arthmetical{
7     public static void main(String[] args) {
8         int a, b, c, sum, prod, avg, bigger;
9         Scanner put = new Scanner(System.in);
10        System.out.println("Enter the numbers :");
11        a = put.nextInt();
12        b = put.nextInt();
13        c = put.nextInt();
14        sum = a + b + c;
15        avg = sum / 3;
16        prod = a*b*c;
17        // using if of indentation (has no curl braces)
18        if (a > b)
19            if (a > c)
20                bigger = a;
21            else
22                bigger = c;
23        else
24            if (b > c)
25                bigger = b;
26            else
27                bigger = c;
28
29        System.out.println("The sum of three integer is " + sum);
30        System.out.println("The average of three integer is " + avg);
31        System.out.println("The product of three integer is " + prod);
32        System.out.println("The bigger of three integer is " + bigger);
33    }
34 }
```

Problem 05. Write an application that asks user to enter the radius of a circle, calculates the area and outputs the result.

Option 01. Using user pi.

```
Circle.java X CircleC.java CircleB.java
Solutions OOP > problem 05 > Circle.java
5
6 import java.util.Scanner;
7
8 public class Circle{
9     public static void main(String[] args) {
10        int r;
11        double area;
12        Scanner put = new Scanner(System.in);
13        System.out.print("Enter the radius of circle :");
14        r = put.nextInt();
15        area = 3.14*r*r;
16
17        System.out.println("The area of a circle is " + area);
18    }
19 }
```

Option 02. Using built-in standard pi.

```
Circle.java  CircleC.java X  CircleB.java
Solutions OOP > problem 05 > CircleC.java
5
6 import java.util.Scanner;
7 import java.lang.Math.*;
8
9 public class CircleC{
10     public static void main(String[] args) {
11         int r;
12         double area;
13         Scanner put = new Scanner(System.in);
14         System.out.print("Enter the radius of circle :");
15         r = put.nextInt();
16         area = Math.PI*Math.pow(r, 2);
17
18         System.out.println("The area of a circle is " + area);
19     }
20 }
```

Topic: Control Statements.

Problem 06. Write a Java application that checks whether the number entered by user is an odd.

```
Odd.java X
Solutions OOP > problem 06 > Odd.java
1 /**
2  * - @author T.cruiz.
3  * - @since Nov-`22.
4  */
5
6 import java.util.Scanner;
7
8 public class Odd{
9     public static void main(String[] args) {
10         int no, ans, n;
11         Scanner scan = new Scanner(System.in);
12         System.out.print("Enter a number :");
13         no = scan.nextInt();
14
15         ans = no % 2;
16         if (ans == 1)
17             System.out.println("The number " + no + " is an odd");
18         else
19             System.out.println("The number " + no + " is not an odd");
20     }
21 }
```

Problem 07. Write a Java application that divides two numbers but checks whether the divisor is not zero.

```
Division.java X
Solutions OOP > problem 07 > Division.java

5
6 import java.util.Scanner;
7
8 public class Division {
9     public static void main(String[] args) {
10         int no1, no2, ans;
11         Scanner scan = new Scanner(System.in);
12         System.out.print("Enter a number :");
13         no1 = scan.nextInt();
14         System.out.print("Enter a devidend :");
15         no2 = scan.nextInt();
16         if (no2 == 0 ){
17             System.out.println("This calculation cant be done!");
18         }else{
19             ans = no1 / no2;
20             System.out.println("The answer is : " + ans);
21         }
22     }
23 }
```

Problem 08. Write a Java application that displays even numbers between 1 and 100.

```
Even.java X
Solutions OOP > problem 08 > Even.java

1 /**
2  * @author T.cruiz.
3  * @since Nov `22.
4  */
5
6 public class Even {
7     public static void main(String[] args) {
8         System.out.print("The even Numbers btn 1 nd 100 are : ");
9         for(int d = 1; d < 100; d++){
10             if(d % 2 == 0){
11                 System.out.print(d + " ");
12             }
13         }
14     }
15 }
```

Problem 09. Write a Java application that calculates the sum of the first 200 counting integers.

```
SumOf200.java X
Solutions OOP > problem 09 > SumOf200.java
1  /**
2   * @author T.cruiz.
3   * @since Nov `22.
4   */
5
6  public class SumOf200 {
7      public static void main(String[] args) {
8          int sum = 0;
9
10         for(int d = 1; d <= 200; d++){
11             sum+=d;
12         }
13
14         System.out.print("The sum of first 200 counting integers is : " + sum);
15     }
16 }
```

Problem 10. Write a Java application that uses looping to print the following table of values:

```
LoopTable.java X
Solutions OOP > problem 10 > LoopTable.java
5
6  public class LoopTable {
7      public static void main(String[] args) {
8
9          System.out.println("N\t10*N\t100*N\t1000*N\n"); // \t == TAB == 4 space (US standard keyboard).
10
11         for(int d = 1; d <= 5; d++){
12             System.out.println(d + "\t" + (d*10) + "\t" + (100*d) + "\t" + (d*1000) );
13         }
14     }
15 }
16 }
```

Topic: Arrays.

Problem 11.

Write an application that initializes all array elements to 10.00.

```
ArrayInit.java X
Solutions OOP > problem 11 > ArrayInit.java
5
6  public class ArrayInit {
7      public static void main(String[] args) {
8
9          double[] arr = new double[20];
10
11         for (int d = 0; d < arr.length; d++){
12             arr[d] = 10.00;
13         }
14     }
15 }
16 }
17 }
```

Problem 12.

Write Java application that obtains data from keyboard and write into an array.


```
ArrayWriter.java X
Solutions OOP > problem 12 > ArrayWriter.java
5  import java.util.Scanner;
6
7  public class ArrayWriter {
8      public static void main(String[] args) {
9
10         double[] arr = new double[20];
11         Scanner cin = new Scanner(System.in);
12
13         for (int d = 0; d < arr.length; d++){
14             System.out.print("enter a value of arr[" + d + "] : ");
15             arr[d] = cin.nextDouble();
16         }
17         /* if you want to print the output uncomment the lines below */
18         // for(int i = 0; i < arr.length; i++){
19         //     System.out.println(" arr[" + i + "] = " + arr[i]);
20         // }
21     }
22 }
```

Problem 13.

Write Java application that displays all elements of an array.

```
ArrayWriter.java  ArrayDisplay.java X
Solutions OOP > problem 13 > ArrayDisplay.java
5
6  public class ArrayDisplay {
7      public static void main(String[] args) {
8
9         double[] arr = {10, 38, 37, 28, 18, 57, 36, 58, 17, 48, 27, 58};
10
11         for(int i = 0; i < arr.length; i++){
12             System.out.println(" arr[" + i + "] = " + arr[i]);
13         }
14     }
15 }
```

Problem 14.

Write Java application that calculates and displays the sum and average of all elements of an array.

```
ArrayArithmetic.java X
Solutions OOP > problem 14 > ArrayArithmetic.java
5
6  public class ArrayArithmetic {
7      public static void main(String[] args) {
8          int sum = 0, avg;
9
10         int[] arr = {10, 38, 37, 28, 18, 57, 36, 58, 17, 48, 27, 58};
11
12         for(int i = 0; i < arr.length; i++){
13             sum += arr[i];
14         }
15
16         avg = sum / arr.length;
17
18         System.out.println(" The Summation of array element is : " + sum);
19         System.out.println(" The average is : " + avg);
20     }
21 }
```

Problem 15. If the float array numbers is initialized as follows: float [] numbers = {50, 10, 20, 60, 120, 90}; Write an application that displays the smallest and largest elements in the float array numbers.

Option 01. Using ternary. (Change array elements)

```
ArrayInspector.java X
Solutions OOP > problem 15 > ArrayInspector.java
6
7 public class ArrayInspector {
8     public static void main(String[] args) {
9
10         float[] numbers = {50, 3, 20, 60, 120, 80, 225};
11
12         float lowFlag = 0, upperFlag = 0;
13
14         for (int c = 0; c < numbers.length; c++){
15             lowFlag = (c == 0) ? numbers[c] : ((lowFlag < numbers[c]) ? lowFlag : numbers[c]);
16             upperFlag = (c == 0) ? numbers[c] : ((upperFlag < numbers[c]) ? numbers[c] : upperFlag);
17         }
18
19         System.out.println("lowflag : " + lowFlag);
20         System.out.println("upperFlag : " + upperFlag);
21     }
22 }
23 }
```

Option 02. Using if-else. (Change array elements)

```
ArrayInspectorIf.java X
Solutions OOP > problem 15 > ArrayInspectorIf.java
5
6 public class ArrayInspectorIf {
7     public static void main(String[] args) {
8
9         float[] numbers = {10, 6, 70, 1240, 10, 60, 120, 80, 565};
10        float lowFlag = 0, upperFlag = 0;
11
12        for (int c = 0; c < numbers.length; c++){
13            if (c == 0){
14                lowFlag = numbers[c];
15                upperFlag = numbers[c];
16            }else{
17                if (lowFlag < numbers[c]){
18                    lowFlag = numbers[c];
19                }else{
20                    lowFlag = lowFlag;
21                }
22
23                if (upperFlag < numbers[c]){
24                    upperFlag = numbers[c];
25                }else{
26                    upperFlag = upperFlag;
27                }
28            }
29        }
30
31        System.out.println("upperFlag : " + upperFlag);
32        System.out.println("lowFlag : " + lowFlag);
33    }
34 }
```

Note.

The following problems have numeral options/ ways to solve as well as compiling them, I have try some, as you will see on code's comments. What is important to consider from here, is a way to compile a package with **parent, model class definition**: On my side I have use a Windows cmd to compile the package as commented on a source-code files.

Eg. On a cmd directory of your project like(C:\users\user\project\>)

```
C:\Users\User\project>javac FileName.java -d .  
C:\Users\User\project>javac FileNameTest.java  
C:\Users\User\project>java FileNameTest
```

Topic: Classes and objects.

Problem 16. Create a class called Box that includes three pieces of information as instance variables - a length (type double), a width (type double) and a depth (type double). Your class should have a constructor that initializes the three instance variables. Provide a method called calculateVolume for calculating the volume of the box. Write a test application named BoxTest that demonstrates class Box's capabilities.

Option 01. In a Single file.

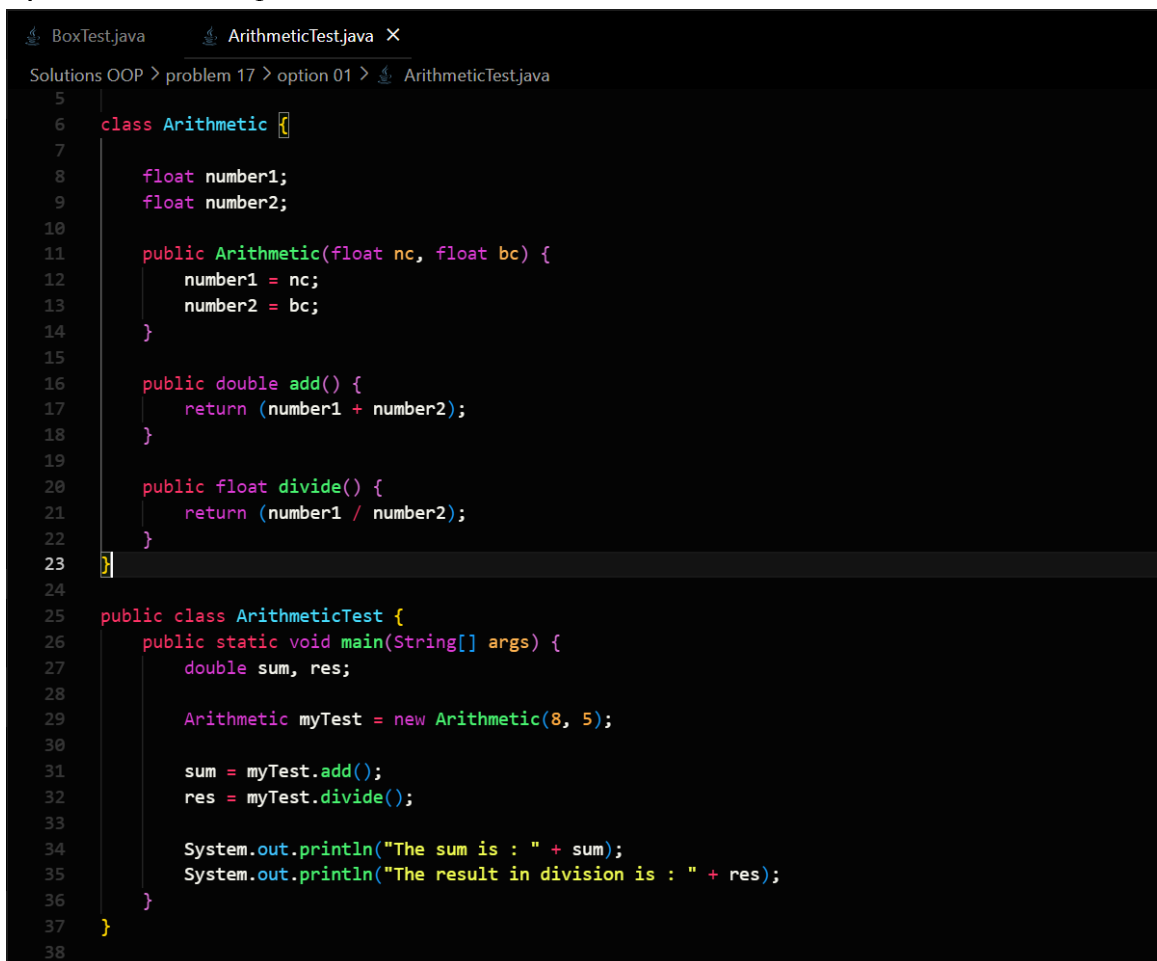


```
BoxTest.java X  
Solutions OOP > problem 16 > Option 01 > BoxTest.java  
5  
6 class Box{  
7     double length;  
8     double width;  
9     double height;  
10  
11     public Box(double h, double w, double l) {  
12         height = h;  
13         width = w;  
14         length = l;  
15     }  
16  
17     public void calculateVolume() {  
18         System.out.println("The volume of a box is : " + (height*width*length));  
19     }  
20 }  
21  
22 public class BoxTest {  
23     public static void main(String[] args) {  
24         Box pcBox = new Box(5, 4, 7);  
25         pcBox.calculateVolume();  
26     }  
27 }
```

Option 02. In a separate files << [Download source-code](#) >>

Problem 17. Create a class called Arithmetic that includes two pieces of information as instance variables - a number1 (type float) and number2 (type float). Your class should have a constructor that initializes the two instance variables. Provide a method called add that calculates the sum of two numbers. Provide another method called divide which calculates the quotient of the two numbers. Write a test application named ArithmeticTest that demonstrates class Arithmetic's capabilities.

Option 01. In a Single file.



```
5
6 class Arithmetic {
7
8     float number1;
9     float number2;
10
11     public Arithmetic(float nc, float bc) {
12         number1 = nc;
13         number2 = bc;
14     }
15
16     public double add() {
17         return (number1 + number2);
18     }
19
20     public float divide() {
21         return (number1 / number2);
22     }
23 }
24
25 public class ArithmeticTest {
26     public static void main(String[] args) {
27         double sum, res;
28
29         Arithmetic myTest = new Arithmetic(8, 5);
30
31         sum = myTest.add();
32         res = myTest.divide();
33
34         System.out.println("The sum is : " + sum);
35         System.out.println("The result in division is : " + res);
36     }
37 }
38
```

Option 02. In a separate files << [Download source-code](#) >>

Problem 18. Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables - a part number (type String), a part description (type String), a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a method named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a double value. If the quantity is not positive, it should be set to 0. If

the price per item is not positive, it should be set to 0.0. Write a test application named InvoiceTest that demonstrates class Invoice's capabilities.

Option 01. In a single file.

```
InvoiceTest.java X
Solutions OOP > problem 18 > option 01 > InvoiceTest.java

5
6 class Invoice{
7
8     String partNo;
9     String partDesc;
10    int itemQuantity;
11    float itemPriceEach;
12
13    /**
14     * @constructor
15     */
16    public Invoice(String pn, String pd, int iq, float ip) {
17        partNo = pn;
18        partDesc = pd;
19        itemQuantity = iq;
20        itemPriceEach = ip;
21    }
22
23    /**
24     * @methods for part number.
25     */
26    public String getPartNumber() {
27        return partNo;
28    }
29    public void setPartNumber(String num) {
30        partNo = num;
31    }
32
33    /**
34     * @methods for part descriptions.
35     */
36    public String getPartDesc() {
37        return partDesc;
38    }
39    public void setPartDesc(String desc) {
40        partDesc = desc;
41    }
42
43    /**
44     * @methods for item quantity.
45     */
46    public int getItemQuantity() {
47        return itemQuantity;
48    }
49    public void setItemQuantity(int quantity) {
50        itemQuantity = quantity;
51    }
52
53    /**
54     * @methods for item price.
55     */
56    public double getItemPrice() {
57        return itemPriceEach;
58    }
59    public void setItemPrice(float price) {
60        itemPriceEach = price;
61    }
62    }
```

```

63
64     /**
65      * @method get the total amount.
66      */
67     public double getInvoiceAmount() {
68         if ((itemQuantity*itemPriceEach) < 0)
69             return 0;
70         else
71             return (itemQuantity*itemPriceEach);
72     }
73 }
74
75 public class InvoiceTest {
76     public static void main(String[] args) {
77
78         Invoice profoma = new Invoice("1001", "For travel", 12, 3);
79
80         System.out.println("Invoice Part No : " + profoma.getPartNumber());
81         System.out.println("Invoice Part Description : " + profoma.getPartDesc());
82         System.out.println("Invoice Item Quantity : " + profoma.getItemQuantity());
83         System.out.println("Item price per each : " + profoma.getItemPrice());
84         System.out.println("Invoice Grand total : " + profoma.getInvoiceAmount());
85     }
86 }
87
88

```

Option 02. In a separate files << [Download source-code](#) >>

Topic: Inheritance.

Problem 19. Create a class called Box that includes three pieces of information as instance variables - a length (type float), a width (type float) and a depth (type float). Your class should have a constructor that initializes the three instance variables. Provide a method called calculateVolume for calculating the volume of the box. Create another class called ColouredBox inherits from class Box. The class ColouredBox includes one more piece of information as its instance variable - colour (type String). Write a test application named ColouredBoxTest that demonstrates class ColouredBox's capabilities.

Option 01. In a single file.

```
ColouredBoxTest.java X
Solutions OOP > problem 19 > option 01 > ColouredBoxTest.java
6  abstract class Box{
7
8      protected float length;
9      protected float width;
10     protected float height;
11
12     public Box(float h, float w, float l) {
13         height = h;
14         width = w;
15         length = l;
16     }
17
18     // an abstract method can be defined to its 1st successor classes || it is like a prototype
19     // any class that extends this(Box) class will have to define it.
20     public abstract double calculateVolume();
21
22 }
23
24 class ColouredBox extends Box {
25
26     protected String colour;
27
28     public ColouredBox(String bxC, float hg, float wd, float ln) {
29         super(hg, wd, ln);
30         colour = bxC;
31     }
32     public String getColour(){
33         return colour;
34     }
35
36     public String setColour(Box color){
37         this.colour = color;
38     }
39
40     // extends a prototype from Box class
41     public double calculateVolume() {
42         return (height*width*length);
43     }
44 }
45
46
47 public class ColouredBoxTest {
48     public static void main(String[] args) {
49         ColouredBox tank = new ColouredBox("Red", 10, 7, 2);
50         System.out.println("Box color : " + tank.getColour() + ", Volume = " + tank.calculateVolume());
51     }
52 }
53
54
55
```

Option 02. In a separate file << [Download source-code](#) >>