

Scenario: Developers need to develop a tax calculator for calculating tax using arithmetic, relational and logical operators.

Problem Statement 1: Declaring instance variables.

Develop a class "TaxCalculator" inside a package "com.example.tax". The class should have an instance float variable named "basicSalary" and a boolean variable named "citizenship".

Problem Statement 2: Usage of Arithmetic operators.

In the TaxCalculator class create a method named calculateTax() that should calculate and print the tax, the tax should be calculated as follows

$$\text{tax} = 30 * \text{basic salary} / 100$$

NOTE: The value stored in the instance variable "basicSalary" should be used in the above calculation.

The calculated tax needs to be stored in another instance float variable "tax".

This method will display the following message in the console.

"The Tax of the employee for the <basic Salary> is <tax>"

Problem Statement 3: How to type cast?

In the TaxCalculator class create a method named deductTax() method this should reduce the tax calculated in problem statement 1 from the basic salary and store it in a instance int variable named "nettSalary".

NOTE: Ensure to use casting for converting float to int.

This method will also display the following message in the console.

"The nett salary of the employee" <nettSalary>

Problem Statement 4: Usage of relational operator

In the TaxCalculator class create a method named validateSalary() method should display a message as below if the basicSalary is greater than 1 lakh and citizenship is true.

"The salary and citizenship eligibility: " <response>

<response> - The value would be printed as true if basic salary > 1 lakh and citizenship is true.

The value would be printed as false for other conditions.

Problem Statement 5: Execute the methods

Develop another class "EmployeeTax" inside the package "com.example.tax" add a main method which sets the values and invoke the following methods in the TaxCalculator object.

Test Case 1: Specify the following values and run the main method

1. Set the value of citizenship as true, basicSalary as 25000.

2. Invoke the methods calculateTax(), deductTax(), validateTax().

Output: The following messages should be displayed in the console

The Tax of the employee for the 25000 is 7500

The nett salary of the employee 17500

The salary and citizenship eligibility: false

Test Case 2: Specify the following values and run the main method

1. Set the value of citizenship as true, basicSalary as 125000.

2. Invoke the methods calculateTax(), deductTax(), validateTax().

Output: The following messages should be displayed in the console

The Tax of the employee for the 125000 is 37500

The nett salary of the employee 87500

The salary and citizenship eligibility: true

Hands-on Exercise Objective

After completing the hands-on exercises, you will be able to:

Develop simple Java program using if-else if - else statement.
Problem Statement:

Develop classes named "Circle", "Square", and "Triangle". Each class should have a method named calculateArea() which calculates the area based on the below logic and display the appropriate message

Develop a class "Shapes" inside a package "com.example.shapes". The class should have one instance variables of type int named "numberOfSides" and a method called calculateShapeArea(). This method takes two arguments one for identifying the shape and other for side length.

This method should identify the shape based on the numberOfSides and calculate the area and print it in console as follows

If numberOfSides is 1 then invoke the "calculateArea" of the Circle Object.

This method will also display the following message in the console.

"The Area of the Circle is"<area>

If numberOfSides is 3 then invoke the "calculateArea" of the Triangle Object.

This method will also display the following message in the console.

"The Area of the Triangle is"<area>

If numberOfSides is 4 then invoke the "calculateArea" of the Square Object.

This method will also display the following message in the console.

"The Area of the Square is"<area>

If numberOfSides is other than the specified ones then display the message "No Shapes Present".

Note: <area> - The value would be printed based on the above logics of different shapes such as Circle, Triangle and Square.

In the Shapes class add a main method which sets the values and invoke the following methods in the Shapes object.

Test Case 1: Specify the following values and run the main method

1. Set the value of numberOfSides as 3 and sideLength as 12.

2. Invoke the method calculateShapeArea ().

Expected Output: The following messages should be displayed in the console

The Area of the Triangle is 62.3538

Test Case 2: Specify the following values and run the main method

1. Set the value of numberOfSides as 4 and sideLength as 15.

2. Invoke the method calculateShapeArea ().

Expected Output: The following messages should be displayed in the console

The Area of the Square is 225

Test Case 3: Specify the following values and run the main method

1. Set the value of numberOfSides as 5 and sideLength as 15.

2. Invoke the method calculateShapeArea ().

Expected Output: The following messages should be displayed in the console

No Shapes Present

Hands-on Exercise Objective

After completing the hands-on exercises, you will be able to:

Develop simple Java program using Switch/Case statement.

Problem Statement:

In the previous activity related to calculate area of circle/triangle and square replace the IF statements with Switch case.

Now Execute the class for the following test cases,

Test Case 1: Specify the following values and run the main method

1. Set the value of numberOfSides as 3 and sideLength as 12.

2. Invoke the method calculateShapeArea ().

Expected Output: The following messages should be displayed in the console

The Area of the Triangle is 62.3538

Test Case 2: Specify the following values and run the main method

1. Set the value of numberOfSides as 4 and sideLength as 15.

2. Invoke the method calculateShapeArea ().

Expected Output: The following messages should be displayed in the console

The Area of the Square is 225

Test Case 3: Specify the following values and run the main method

1. Set the value of numberOfSides as 5 and sideLength as 15.

2. Invoke the method calculateShapeArea().

Expected Output: The following messages should be displayed in the console

No Shapes Present

Hands on Exercise Objective

After completing the hands-on exercises, you will be able to:

Develop simple Java program using nested IF statement.

Problem Statement:

Develop a calculator for calculating the simple interest for a principle, number of years and rate of interest.

$$\text{Simple interest} = P * N * R/100.$$

Create a class SimpleInterestCalculator with three instance variable double principal amount, int number of years. Develop a method calculateSimpleInterest() and implement the logic as follows.

The business logic for calculating the simple interest is as follows.

If principal is greater than one lakh

If number of years is greater than 10

Simple interest = $P * N * 10/100$

Otherwise if number of years < 10

Simple interest = $P * N * 9.5/100$

Else if amount is less than one lakh

If number of years is greater than 10

Simple interest = $P * N * 5/100$

Otherwise if number of years < 10

Simple interest = $P * N * 4.5/100$

The calculated simple interest is to be printed in the calculateSimpleInterest() method.

"The interest amount for a principal of <principal amount> and years <number of years> is <Result>"

Develop a main method which sets the instance variables and invoke the method calculateSimpleInterest().

Now Execute the class for the following test cases,

Test Case 1: Specify the following values and run the main method

1. Set the value of principal amount as 2 lakh and number of years has 12

2. Invoke the method calculateSimpleInterest ().

Expected Output: The following messages should be displayed in the console

The interest amount for a principal 200000 and years 12 is 2400000

Test Case 2: Specify the following values and run the main method

1. Set the value of principal amount as 50,000 and number of years has 12

2. Invoke the method calculateSimpleInterest ().

Expected Output: The following messages should be displayed in the console

The interest amount for a principal 50000 and years 12 is 30000

Test Case 3: Specify the following values and run the main method

1. Set the value of principal amount as 50,000 and number of years has 5

2. Invoke the method calculateSimpleInterest ().

Expected Output: The following messages should be displayed in the console

The interest amount for a principal 50,000 and years 5 is 11,250.

Hands on Exercise Objective

After completing the hands-on exercises, you will be able to:

Develop simple Java program using For loops and nested for loops statement.

Problem Statement # 1:

Develop a program which prints all the odd numbers between 1 and 200.

The output should be printed in this format

1,3,5.....goes on till it prints the last odd number

Problem Statement # 2:

Develop a program to print the following format

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Hint: Use nested for loops to achieve this.

Hands on Exercise Objective

After completing the hands-on exercises, you will be able to:

Develop simple Java program using while statement

Problem Statement # 1:

Develop a program which prints the prime numbers between 1 and 150

The output should be printed in this format

1,3,5.....goes on till it prints the last prime number.

Prime number definition: A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself. For example 5 is prime, as only 1 and 5 divide it.

Problem Statement 1: Declaring and using constructors

Create a class Circle.java in a package "com.example.shapes" , add a float instance variable radius and add a default constructor (Constructor 1) for the class. This constructor should initialize the radius to a default value 1.5f.

The above constructor should be invoked from a main method from another class, Shape.java (in different package com.example.geometry).

Problem Statement 2: Overloading constructors and using "this" keyword.

In the Circle.java class created above add an instance float variable pi and create two overloaded constructors.

Constructor 2- with a float argument name radius. The constructor should initialize the class variable radius with the method argument radius.

NOTE: The instance variable and the method argument should be named same as "radius".

Constructor 3- with two float arguments radius and pi. Default the class pi value to 3.5 and set the instance variable with the radius method argument.

The constructor "constructor 2" should be invoked from a main method from class, Area.java (in a package com.example.shapes).

Problem Statement 3: Constructor Chaining

In Circle.java, invoke the Constructor 3 created in the previous step from Constructor 2.

Problem Statement 4: Applying access specifiers to constructors/variables

- a. (Other classes must not be able to call this constructor). Also restrict the access to the variable radius to class level
- b. Provide package level access to Constructor 2 (Classes in other package must not be able to access this constructor). Also provide package level access to the variable pi.

Problem Statement 5: Create two methods and calculate area and circumference of a Circle

In the Circle.java class, create two methods as listed below

- a. Method 1 - calculateCircleArea should accept the float radius as parameter and calculate the area ($\pi * r * r$). It should return the result value to the main method where it should be printed in the console.
- b. Method 2 - calculateCircumference should accept float radius as parameter and calculate the circumference ($2 * \pi * r$). It should return the result value to the main method where it should be printed in the console.

Call these two methods from the main method in Circle.java by passing appropriate parameters.

Activity on Overloading

This activity is to find out more on Overloading. There are 5 pairs of methods given.

Create a class named Testing. Add the following pair of methods one by one and find out which two methods can co-exist (overloaded). Mention the result (Yes/No) below.

1.

- a. `int add(int num1, int num2)`
- b. `void add(int num1, int num2)`

2.

- a. `int add(int num1, float num2, int num3)`
- b. `int add(int num4, float num5, int num6)`

3. Assumption : Triangle is a derived class of Shape class. You need to make both the classes before trying out this activity.

- a. `Shape draw(int x, int y)`
- b. `Triangle draw(int x, int y)`

4.

- a. `Integer add(int num1, int num2)`
- b. `Long add(int num1, int num2)`

5.

- a. `public Long draw(long x, long y)`
- b. `public Integer draw(Integer x, Integer y)`

Activity on Constructor Overloading

Refer to the code given for Constructor Overloading. There is a Class named Employee which does not have any Constructors.

```
package com.example;
public class Employee {
    private String empName;
    private String empAddress;
    private int empId;

    public String getEmpName() {
        return empName;
    }
    public void setEmpName(String empName) {
        this.empName = empName;
    }
    public String getEmpAddress() {
        return empAddress;
    }
    public void setEmpAddress(String empAddress) {
        this.empAddress = empAddress;
    }
    public int getEmpId() {
        return empId;
    }
}
```

```

}
public void setEmpId(int empId) {
this.empId = empId;
}
}

```

In the Client code (MyMain.java), we are creating the objects in multiple ways.

```

package com.example;
public class MyMain {
public static void main(String[] args) {
Employee empObj1 = new Employee();
Employee empObj2 = new Employee("Rama" , 101);
Employee empObj3 = new Employee("Rama" , 101 , "Bangalore");
Employee empObj4 = new Employee(101, "Rama");
}
}

```

Define the appropriate Constructors for the Employee Class so that the above code compiles.

Activity on Overloading

This activity is to find out more on Overloading. There are 5 pairs of methods given.

Create a class named Testing. Add the following pair of methods one by one and find out which two methods can co-exist (overloaded). Mention the result (Yes/No) below.

1.

- a. int add(int num1, int num2)
- b. void add(int num1, int num2)

2.

- a. int add(int num1, float num2, int num3)
- b. int add(int num4, float num5, int num6)

3. Assumption : Triangle is a derived class of Shape class. You need to make both the classes before trying out this activity.

- a. Shape draw(int x, int y)
- b. Triangle draw(int x, int y)

4.

- a. Integer add(int num1, int num2)
- b. Long add(int num1, int num2)

5.

- a. public Long draw(long x, long y)
- b. public Integer draw(Integer x, Integer y)

Activity on Constructor Overloading

Refer to the code given for Constructor Overloading. There is a Class

named Employee which does not have any Constructors.

```
package com.example;
public class Employee {
    private String empName;
    private String empAddress;
    private int empId;

    public String getEmpName() {
        return empName;
    }
    public void setEmpName(String empName) {
        this.empName = empName;
    }
    public String getEmpAddress() {
        return empAddress;
    }
    public void setEmpAddress(String empAddress) {
        this.empAddress = empAddress;
    }
    public int getEmpId() {
        return empId;
    }
    public void setEmpId(int empId) {
        this.empId = empId;
    }
}
```

In the Client code (MyMain.java), we are creating the objects in multiple ways.

```
package com.example;
public class MyMain {
    public static void main(String[] args) {
        Employee empObj1 = new Employee();
        Employee empObj2 = new Employee("Rama" , 101);
        Employee empObj3 = new Employee("Rama" , 101 , "Bangalore");
        Employee empObj4 = new Employee(101, "Rama");
    }
}
```

Define the appropriate Constructors for the Employee Class so that the above code compiles.

What is the expected output?

```
public class Profile
{
    private Profile(int w)
    {
        System.out.println(w);
    }
    public static Profile()
    {
```



```

    System.out.println(10);
}
public static void main(String args[])
{
    Profile obj = new Profile(50);
}
}

```

What is the result of compiling and running the following code?

```

public class Test
{
    static int p = test();
    static public int test()
    {
        System.out.println(p);
        return 99;
    }
    public static void main(String[] args)
    {
        System.out.println(p);
    }
}

```

What is the result of compiling and running the following code?

```

public class Tester
{
    static int x = 4;
    public Tester()
    {
        System.out.print(this.x);
        Tester();
    }
    public static void Tester()
    {
        System.out.print(this.x);
    }
    public static void main(String... args)
    {
        new Tester();
    }
}

```

What is the output?

```

class InitDemo
{
    static int i=demo();
    static
    {
        System.out.println(i);
    }
    InitDemo()
    {
        System.out.println("hello 1");
    }
    public static void main(String... args)
    {

```

```

System.out.println("Hello 2");
}
static int demo()
{
    System.out.println("Inside Demo");
    return 10;
}
}
What is the output?
class InitDemo{
    int a;
    public InitDemo(int val)
    {
        a=val;
        System.out.println("Value of  a:"+a);
    }
}
class MainClass
{
    public static void main(String... args)
    {
        new InitDemo();
    }
}

```

Create a class that calculates the factorial of number
 Class Name Factorial Method Name calculateFactorial Method Description
 Calculates the factorial of a number Argument int n Return Type int
 - Factorial of a number Logic Factorial of a number is given by N
 $\text{factorial} = n * (n-1) * (n-2) \dots 1$

Create a class that returns the biggest number.
 Class Name GreaterFinder
 Method Name findGreater2
 Method Description Finds the biggest of two numbers
 Argument int number1,number 2
 Return Type int - biggest number
 Logic The bigger of the two numbers should be determined using a
 single operator.
 Method Name findGreater2
 Method Description Finds the biggest of three numbers
 Argument int number1,number 2,number3
 Return Type int - biggest number Logic The biggest of three numbers
 using if statement.

What is the output of running the following code?

```

public class Tester
{
    public static void main(String[] args)
    {
        boolean flag = true;
        int x = 5;
        int result = (8 * 2) % x;
        System.out.print(!(result > 0) ^ true ^ (flag = false));
        System.out.print(flag);
    }
}

```

What is the output of running the following code?

```
public static void main(String[] args)
{
    int x = 0, y = 0; x = 5 + y++;
    System.out.print(x + " " + y);
    x = 0; y = 0; x = 5 + ++y;
    System.out.print(x + " " + y);
}
```

What is the result of compiling and running the following code?

```
public class Test {
    public static void main(String... args) {
        int a =5 , b=6, c =7;
        System.out.println("Value is "+ b +c);
        System.out.println(a + b +c);
        System.out.println("String "+(b+c));
    }
}
```

What is the result of compiling and running the following code?

```
public class Test {
    public static void main() {
        System.out.println("hello");
    }
}
```

Create a class with a method which can accept a String name and print a message "hello<name>"

Class Name MessagePrinter

Method Name printMessage

Method Description Prints the message Argument String name

Return Type Void

Logic Print the message using the console output command.

Create a class which contains methods for finding the sum and difference of two numbers

Class Name Calculator

Method Name calculateSum

Method Description Calculates the sum of two numbers

Argument int number1, int number2

Return Type int - Sum

Logic Calculate the sum of the two numbers number1 nad number2 and return the sum.

Method Name calculateDifference

Method Description Calculates the difference between two numbers

Argument int number1 , int number2

Return Type int -

difference Logic Calculate the difference between the numbers number1 and number2 and return the difference.

What is the result of compiling and running the following code?

2. What is the result of compiling and running the following code?

```
public class Tester
{
    public static void main(String[] args)
    { int x = 0; int y = 0;
      if ((true & false) | (true & false) & x++>0)
```

```

System.out.print("stmt1 ");
if (false || true | (++y> 0 | y++>0))
System.out.print("stmt2 ");
System.out.print(x+" "+y);
}
}
public static void main(String[] args)
{
int x = 1, y = 0;
if ((y == x++) && (x < ++y))
{
}
System.out.println(x + " " + y);
}

```

4. What is the result of compiling and running the following code?

```

public static void main(String[] args)
{
int x = 0, y = 0;
if ((y == x++) | (x < ++y))
{ ++y;
}
System.out.println(x + " " + y);
}

```

What is the result of compiling and running the following code?

```

public class Tester
{
public static void main(String[] args)
{
do
{
System.out.print("inside do");
}
while (false);
while (false)
{
System.out.print("inside while");
}
System.out.print("outside");
}
}

```

What is the result of compiling and running the following code?

```

public static void main(String[] args)
{
int j = 10;
switch (1)
{
case 20: j += 1;
case 40: j += 2;
default: j += 3;
case 0: j += 4;
}
System.out.print(j);
}

```

Knowing that character 'a' has a numeric value equal to 97, what is the output of the following code

```
public class CaseTest
{
public static void main(String... s)
{ char c = 'a';
  switch (c)
  {
case 'a':
{
System.out.println("a");
break;
}
case 97:
{
System.out.println("97");
break;
}
}
}
```

What will be the output of the program?

```
class Test
{
    public static void main(String [] args)
    {
        int x= 0;
        int y= 0;
        for (int z = 0; z < 5; z++)
        {
            if (( ++x > 2 ) && (++y > 2))
            {
                x++;
            }
        }
        System.out.println(x + " " + y);
    }
}
```

What is the result of compiling and running the following code?

```
public class Tester { boolean isEven(int x) { return (x % 2 == 0) ?
true : false; } public static void main(String[] args) {
System.out.print(isEven(2)); System.out.print(isEven(3));
System.out.print(isEven(4)); } }
```

What will be the output on running the code?

```
public static void main(String[] args) { String entries[] =
{"entry1","entry2"}; int count=0; while (entries [count++]!=null){
System.out.println(count); } System.out.println(count); }
```

What is the output?

```
class InitDemo{ static int i=demo(); static{System.out.println(i);}
InitDemo(){ System.out.println("hello 1"); } public static void
main(String... args){ System.out.println("Hello 2"); } static int
demo(){ System.out.println("Inside Demo"); return 10; } }
```

What is the result of compiling and running the following code?

```

public class Tester { static int x = 4; int y; public Tester(int a,int
b) { this.x=a; this.y=b; System.out.println("Value of  x :"+this.x);
System.out.println("Value of  y :"+this.y);  Tester(); } public
static void testMethod() {  System.out.print(this.x);  } public
static void main(String... args) {  new Tester(5,10); Test mytest=new
Tester(); Mytest.testMethod(); }
}

```

What is the result of compiling and running the following code?

```

public class Test {      static int p = test();      public static  int
test() {  System.out.println(p);      return 99;  }      public static
void main(String[] args)  {  System.out.println(p);  }  }

```

```

public class LoopTest { public static void main(String args[]) { int a
= 5; outside: for (int i = 1; i < 3; i++) { inside: for (int j = 1; j
< 3; j++) { System.out.print(j); if (a++ >6) continue inside; break
outside; } } } }

```

What is the expected output?

```

public class LoopTest { public static void goLoop(int a) { start: for
(int i = 1; i < 2; i++) { for (int j = 1; j < 2; j++) { if (a > 5) {
break start; } System.out.print(i + j); } } } public static void
main(String args[]) { LoopTest.goLoop(15); } }

```

Create a class that can calculate the sum of the first n numbers of the following series

1+3+5+7+9+11..... and return the sum

Class Name SeriesCalculator

Method Name calculateSum

Method Description Calculates the sum of first n numbers Argument int n Return Type int - Sum

Logic Calculate the sum of the series which represents a arithmetic progression using the formula $Sum = (n/2) * [2*a + (n-1)*d]$

Where n- number of terms

a-First term of the series (Here 1)

d-common difference (Here 2)

Navik Nav <navikdcnotes@gmail.com>

Mon, Jan 20, 1:52 AM (3 days ago)

to vishnusail229

Classes and Objects

1. Create a class called biggest consisting of a single dimensional array of a particular size as it data members and a function called display() in which the largest element of the array is found and displayed.

2. Create a class rectangle. The class has attributes length and width, each of which defaults to 1. It should have set and get functions for both length and width.

It should have member function that calculate the area of the rectangle.

3. Write a class Person with FirstName and LastName as data members. Override the toString() method to return the fullname of the person. Define constructors to take appropriate parameters.

Write a program which keeps track of the number of objects that are created and display the count in a function called display().

5. Create an Employee class with members emp no, name, department and salary. In main create a reference variable of type Employee. Allocate memory for the Employee using new operator. Read the employee details from command line arguments and initialize the data members. Also display the data members values.

Create a class called BankAccount with AcctNo , balance ,AcctType as data members. Create objects to hold specific values of the data members which are initialized using constructors. Write functions to perform the below operations

a) To deposit an amount of 10000

b) To withdraw an amount of 5000

c) To display the AccountDetails

Create a Employee class with members emp no, name, department and salary. Use single dimensional array of employee objects of size 10, to read 10 employees data from command line arguments. After reading display the same data on the console from array. Also have a function to determine which employees salary is the highest

a) Create a class to hold information about books. Write a function to display information about the book.

Data members : Title , Author , cost , no_of_books

b) Create 3 instances of the above class and initialize the members of the class with the data accepted from the user.

c) Accept a title and the no_of books required from the user. Find whether the book exists . If it exists , check if the no_of books are sufficient and if so indicate the total cost the books.

LANGUAGE FUNDAMENTALS - EXERCISES

1. Write a program to accept a double value. Separate the whole value from the fractional value and store them in two variables. Display the same.

2. Write a program to accept three numbers and find the largest of three numbers using ternary operator.
3. Write a program to accept a number and display whether the number is odd or even using the ternary operator.
4. Write a program to accept three numbers and find the largest and second largest values of three numbers
5. Write a program to accept a student's name and three marks. Display the total and average. Display result as First class, Second class, Pass class and Fails.
6. Write a program to accept a number and display the last digit of the number in words.
7. Write a program to accept an alphabet and display whether its a vowel or consonant
8. Write a program to find the sum of all odd numbers from 1 to N. Accept N. Display the sum.
9. Write a program to find the reverse of a number. Store the reverse value in a different variable. Display the reverse.
10. Write a program to find the factorial of a non-negative number.
11. Write a program to generate the following series:
$$1, -2, 3, -4, 5, -6, \dots N$$
12. Write a program to find the sum of all the prime numbers in the range n to m.

Display each prime number and also the final sum.
13. Write a program to store N elements in an array of integer. Display the elements. Accept a number to be searched. Display whether the number is found or not in the array (LINEAR SEARCH).
14. Write a program to find the factorial of a number using recursion.

