

OBJECT ORIENTED PROGRAMMING IN JAVA - EXERCISES

.....

CHAPTER 1

1. Write Text-Based Application using Object-Oriented Approach to display your name.
2. Write a java Applet to display your age.

CHAPTER 2

3. Write a program that calculates and prints the product of three integers.
4. Write a program that converts a Fahrenheit degree to Celsius using the formula:

$$\text{Celsius} = 5/9(\text{fahrenheit}-32)$$

5. Write an application that displays the numbers 1 to 4 on the same line, with each pair of adjacent numbers separated by one space.

Write the application using the following techniques:

- a. Use one System.out.println statement.
 - b. b. Use four System.out.print statements.
 - c. c. Use one System.out.printf statement.
6. Write an application that asks the user to enter two integers, obtains them from the user and prints their sum, product, difference and quotient (division).

CHAPTER 3

7. Write an application that asks the user to enter two integers, obtains them from the user and displays the larger number followed by the words "is larger". If the numbers are equal, print "These numbers are equal"
8. Write an application that inputs three integers from the user and displays the sum, average, product, smallest and largest of the numbers.

9. Write an application that reads two integers, determines whether the first is a multiple of the second and print the result. [Hint Use the remainder operator.]

10. The process of finding the largest value (i.e., the maximum of a group of values) is used frequently in computer applications. For example, a program that determines the winner of a sales contest would input the number of units sold by each sales person. The sales person who sells the most units wins the contest. Write a Java application that inputs a series of 10 integers and determines and prints the largest integer. Your program should use at least the following three variables:

a. counter: A counter to count to 10 (i.e., to keep track of how many numbers have been input and to determine when all 10 numbers have been processed).

b. number: The integer most recently input by the user.

c. largest: The largest number found so far.

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

N	10*N	100*N	1000*N
1	10	100	1000
2	20	200	2000
3	30	300	3000
4	40	400	4000
5	50	500	5000

12. Write a complete Java application to prompt the user for the double radius of a sphere, and call method sphereVolume to calculate and display the volume of the sphere. Use the following statement to calculate the volume:

```
double volume = ( 4.0 / 3.0 ) * Math.PI * Math.pow( radius, 3 )
```

CHAPTER 4

13. Write statements that perform the following one-dimensional-array operations:

- d. Set the 10 elements of integer array counts to zero.
- e. Add one to each of the 15 elements of integer array bonus.
- f. Display the five values of integer array bestScores in column format

14. Write a Java program that reads a string from the keyboard, and outputs the string twice in a row, first all uppercase and next all lowercase. If, for instance, the string "Hello" is given, the output will be "HELLOhello"

15. Write a Java application that allows the user to enter up to 20 integer grades into an array. Stop the loop by typing in -1. Your main method should call an Average method that returns the average of the grades. Use the DecimalFormat class to format the average to 2 decimal places

CHAPTER 5

16. Modify class Account (in the example) to provide a method called debit that withdraws money from an Account. Ensure that the debit amount does not exceed the Account's balance. If it does, the balance should be left unchanged and the method should print a message indicating —Debit amount exceeded account balance. Modify class AccountTest (in the example) to test method debit.

17. 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.

18. Create a class called Employee that includes three pieces of information as instance variables—a first name (typeString), a last name (typeString) and a monthly salary (double). Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10% raise and display each Employee's yearly salary again.

19. Create a class called Date that includes three pieces of information as instance variables—a month (typeint), a day (typeint) and a year (typeint). Your class should have a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes(/). Write a test application named DateTest that demonstrates classDate's capabilities.

CHAPTER 6

20. Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12 this interest should be added to savingsBalance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value. Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of \$2000.00 and \$3000.00,

respectively. Set `annualInterestRate` to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the `annualInterestRate` to 5%, calculate the next month's interest and print the new balances for both savers.

21. Create a class called `Book` to represent a book. A `Book` should include four pieces of information as instance variables-a book name, an ISBN number, an author name and a publisher. Your class should have a constructor that initializes the four instance variables. Provide a mutator method and accessor method (query method) for each instance variable. In addition, provide a method named `getBookInfo` that returns the description of the book as a `String` (the description should include all the information about the book). You should use this keyword in member methods and constructor. Write a test application named `BookTest` to create an array of object for 30 elements for class `Book` to demonstrate the class `Book`'s capabilities

CHAPTER 7

22.

a. Create a super class called `Car`. The `Car` class has the following fields and methods.

- `int speed;`
- `double regularPrice;`
- `String color;`
- `double getSalePrice();`

//filename: `Car.java` //Car class

b. Create a sub class of `Car` class and name it as `Truck`. The `Truck` class has the following fields and methods. ◦ `int weight;`

◦double getSalePrice();//If weight > 2000, 10% discount. Otherwise, 20% discount.

c. Create a subclass of Car class and name it as Ford. The Ford class has the following fields and methods

◦int year;

◦double

getSalePrice();//From the sale price computed from Car class, subtract the manufacturerDiscount.

d. Create a subclass of Car class and name it as Sedan. The Sedan class has the following fields and methods.

◦int length;

◦double

getSalePrice();//If length > 20 feet, 5% discount, Otherwise, 10% discount.

e. Create MyOwnAutoShop class which contains the main() method. Perform the following within the main() method.

Create an instance of Sedan class and initialize all the fields with appropriate values. Use super(...) method in the constructor for initializing the fields of the superclass.

Create two instances of the Ford class and initialize all the fields with appropriate values. Use super(...) method in the constructor for initializing the fields of the super class. www.oumstudents.tk

◦ Create an instance of Car class and initialize all the fields with appropriate values. Display the sale prices of all instance.

CHAPTER 9

23. Write an applet that asks the user to enter two floating-point numbers, obtains the two numbers from the user and draws their sum, product (multiplication), difference and quotient (division). Use the techniques shown in example.

CHAPTER 10

24. Create an applet that can display the following component. No event handling is needed for the components.



A form with three labels and text input fields. The labels are "name", "address", and "e-mail", each followed by a rectangular text input field.

CHAPTER 11

26. Temperature Conversion

a. Write a temperature conversion applet that converts from Fahrenheit to Celsius. The Fahrenheit temperature should be entered from the keyboard (via a JTextField). A JLabel should be used to display the converted temperature. Use the following formula for the conversion: $\text{Celcius} = ((5/9) * (\text{Ferenheit} - 32))$.

b. Enhance the temperature conversion applet of Q1 by adding the Kelvin temperature scale. The applet should also allow the user to make conversions between any two scales. Use the following formula for the conversion between Kelvin and Celsius (in addition to the formulain Q1): $\text{Kelvin} = \text{Celcius} + 273.15$

