

Assignments

Core Java

Basics

1. Convert the celsius temperature input from keyboard into its fahrenheit equivalent using following formula :

$$c=(f-32)/1.8$$

2. Accept the value of money in rupees and convert it into dollars.

3. Determine the sum of the following series (assume $n=10$)

4. Print the fibonacci series on the screen for 10 terms:

0 1 1 2 3 5 8 13

5. Find squares of all the numbers from 1 to 100

6. Accept a number as input and find its sum of digits. ex. 154 : $1+5+4= 10$

7. Input a number and print it in reverse order : ex 12345--->54321

more practice questions

1. Input a long number and count total number of even digits in it.

Exm: 542587545 : 4 even digits

2. Find the sum of all the integers greater than 100 and less than 200 that are divisible by 7

3. Accept marks of 10 different students and analyze it. That is, count total first, second and pass class students

4. print following output using for loop

*

* *

* * *

* * * *

* * * * *

5. Print the diamond shape using *

6. Accept a number and find its binary equivalent.

Looping

Q1. Write a program to calculate overtime pay of 10 employees. Overtime is paid at the rate of Rs. 12.00 per hour for every hour worked above 40 hours. Assume that employees do not work for fractional part of an hour.

Q2. Write a program to find the factorial value of any number entered through the keyboard.

Q3. Two numbers are entered through the keyboard. Write a program to find the value of one number raised to the power of another.

Q4. Write a program to print out all Armstrong numbers between 1 and 500. If sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number. For example, $153 = (1 * 1 * 1) + (5 * 5 * 5) + (3 * 3 * 3)$

Q5. Write a program to enter the numbers till the user wants and at the end it should display the count of positive, negative and zeros entered.

Q6. Write a program to print all prime numbers from 1 to 300. (Hint: Use nested loops, **break** and **continue**)

Q7. Write a program to add first seven terms of the following series..
 $1/1! + 2/2! + 3/3! + \dots$

Q8. Write a program to generate all combinations of 1, 2 and 3.

Q9. Write a program to print the multiplication table of the number entered by the user. The table should be displayed in the following form.

$5 * 1 = 5$

$5 * 2 = 10$

....

....

Q10: A machine is purchased which will produce earning of Rs. 1000 per year while it lasts. The machine's cost is 6000 and will have salvage of rs. 2000 when it is condemned. If 12% per annum is earned on alternate investments, what would be the minimum life of the machine to make it a more attractive investment compare to alternative arrangement.

Arrays

Q1: Write a function `reverselt()`, that reverse a String. Use a for loop that swaps the first and last characters, then second and next-to-last characters and so on. The string should be passed to `reverselt()` method as an argument and it should return the reverse string.

Write a program to exercise `reverselt()`. The program should get the string from the user. Call `reverselt()` and print the output. Check the program with following String "Able was I era I saw elba."

Q2: Create a class called `Employee` that contains a name and `employeeid`. Include the member functions as `getData()` and `putData()` respectively for getting and displaying values.

Write a main class to exercise `Employee` class. it should create the array of type `employee` and then allow the user to input the data for 10 employees, ask user to enter the employee id and display the information of the entered employee id.

Q3: Start with a program that allows the user to input a number of integers, and then store them in an int array. Write a function called `maxInt()` that goes through the array, element by element, looking for the largest one. The function should take as arguments the address of the array and the number of elements in it, and return the index number of the largest element. The program should call this function and then display the largest element and its index number.

Q4: Create a 5 story car parking system simulation, where 1st story will be having the capacity to park 100 cars, 2nd story will be having the capacity of 70 cars, 3rd will be having 50 cars, 4th will be having 20 cars and 5th will be having 10 cars.

Your program should show on each floor how may cars are parked and how much is the empty space.

Q5. Create a two dimensional array for storing salutation and names. Salutations are fixed like Mr. Mrs. Ms., names can vary.

For example

Mr. Mrs. Ms.

Pratik Aishwarya Surya Dishant Somesh Praveen

Your program should display the result as

Mr. Pratik

Ms. Aishwarya

Mrs. Surya

Mr. Dishant

Mr. Somesh

Mr. Praveen

OOPs

Q1: Create an Employee class having name and age as fields and doSomething () as methods. Write a TestEmployee Class to set the values of name and age of the employee and let not user allow to enter age in alphabets or negative age. Report the appropriate error message to user.

Class & Object

Q1: Create a class that imitates part of the functionality of the basic data type int. Call the class Int (note different capitalization). The only data in this class is an int variable. Include member functions to initialize an Int to 0, to initialize it to an int value, to display it (it looks just like an int), and to add two Int values.

Write a program that exercises this class by creating one uninitialized and two initialized Int values, adding the two initialized values and placing the response in the uninitialized value, and then displaying this result.

Q2: Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay a 50 cent toll. Mostly they do, but sometimes a car goes by without paying. The tollbooth keeps track of the number of cars that have gone by, and of the total amount of money collected. Model this tollbooth with a class called TollBooth. The two data items are a type int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both of these to 0. A member function called payingCar() increments the car total and adds 0.50 to the cash total. Another function, called noPayCar(), increments the car total but adds nothing to the cash total. Finally, a member function called display() displays the two totals. Make appropriate member functions const.

Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car. Pushing the Esc key should cause the program to print out the total cars and total cash and then exit.

Q3: Create a class called time that has separate int member data for hours, minutes, and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. Another member function should display it, in 11:59:59 format. The final member function should add two objects of type time passed as arguments.

A main() program should create two initialized time objects (should they be const?) and one that isn't initialized. Then it should add the two initialized values together, leaving the result in the third time variable. Finally it should display the value of this third variable.

Make appropriate member functions const.

Q4: Create an employee class. The member data should comprise an int for storing the employee number and a float for storing the employee's compensation. Member functions should allow the user to enter this data and display it. Write a main() that allows the user to enter data for three employees and display it.

Q5: Create a class of type JoinDate that contains three members: the month, the day of the month, and the year, all of type int. (Or use day-month-year order if you prefer. It should also have two member functions: getdate(), which allows the user to enter a date in 12/31/02 format, and showdate(), which displays the date.

Q6: Extend the employee class of to include a JoinDate class and an etype enum. An object of the JoinDate class should be used to hold the joining date of first employment; that is, the date when the employee was hired. The etype variable should hold the employee's type: laborer, secretary, manager, and so on. These two items will be private member data in the employee definition, just like the employee number and salary. You'll need to extend the getters to obtain this new information from the user and display it. These functions will probably need switch statements to handle the etype variable. Write a main () program that allows the user to enter data for three employee variables and then displays this data.