

Institute of Engineering & Technology, DAVV, Indore
Department of Computer Engineering & Information Technology
Subject Code : 2CO254 / 2IT454 BE II Computer Engg.

Lab Assignment #2

Last date of Submission : 11 February 2012

Note: Assignment Program files should have the same name as their problem serial number. For example for third question in the assignment, A2P3.java file is to be created which should consist of all required classes/code. For any ambiguity or insufficient information regarding solution of problem students are welcome to discuss. Assignment is aimed to have practice on java coding. Students should do the assignment by themselves.

A2P1. Write a program to sort the array first, and then copy elements of array into another using the System.arraycopy () API.

A2P2. Write and run a Java program that capitalizes a two-word name. For example, the input
iEt dAvV would produce the output lEt Davv

A2P3. Write a program that initializes an integer variable n with the value 5814 and then uses the quotient and remainder operators to extract and print each digit of n. The output should look like this: n = 5814 The digits of n are 5, 8, 1 and 4.

A2P4. Write a function names *int hexvalue(String)*. The argument string contains hexadecimal (for example 2a3 or 1b or 39). Function is required to return the equivalent int value. Write main() to test run the method.

A2P5. Redo the above problem (A2P4) to check first whether the argument string contains “hexadecimal number” only. Method should reject string like “23t1”.

A2P6. Define a class for rectangle objects defined by two points, the top-left and bottom-right corners of the rectangle.

- Declare a method which returns a rectangle object that encloses the current object and the rectangle passed as an argument;
- Declare a method which takes an array of rectangle objects and returns a rectangle object which encloses all the rectangle objects of the array.

A2P7. Define a class, *mcmlength*, which represents the length measured in meters, centimeters, and millimeters, each stored as integers. Declare methods to add and subtract length objects. Include methods to enlarge and reduce length object by a factor. Also declare methods which returns product of two length objects and which compares two objects and returns 1 if first object is greater, return 0 if both are equal otherwise -1. Include constructors that accepts

- i. three arguments to be considered as meter,cm, and mm;
- ii. two arguments in cm and mm;
- iii. one argument in mm;
- iv. without arguments, which creates an object with the length 0;
- v. a double argument in meters.

A2P8. Define the class, *Tkgweight*, to represent a weight in tons, Kilograms, and grams, and include a similar range of methods and constructors as the previous example. Demonstrate this class by creating and operating some objects of the class.

A2P9. Define a class *Circle* and write a method that computes the area of a circular region, given the radius of the inner and outer circles.

A2P10. Write the currency converter program. The program should also have the provision of storing(not in disk/file)/modifying the exchange rates. The output may be displayed in tabular form or individual currency conversion.

A2P11. Write an application which does following.

- a. Reads a character and display the ASCII code.
- b. Reads a sentence and prints the sentence with all uppercase letter change to lowercase letters and all lowercase letters into uppercase letters.
- c. Reads a sentence and prints out the sentence in reverse order

A2P12. Write and run a Java program that generates a random double (Use Random class from java.util), determines which quintile of the unit interval it is in, and reports it. A *quintile* is one of the five equal sized pieces of the whole. The quintiles of the unit interval are 0 to 1/5, 1/5 to 2/5, 2/5 to 3/5, 3/5 to 4/5, and 4/5 to 1.

A2P13. Write and run a Java program that generates a random year between 1800 and 2000 and then reports whether it is a leap year. A leap year is an integer that is either divisible by 400 or is divisible by 4 but not 100. To generate an integer in the range 1800 to 2000, use

*int year = Math.round(200*x + 1800);*

where x is a random float. The round () method of the Math class returns the integer nearest the float passed to it. The transformation $y = 200x + 1800$ converts a number in the range $0 \leq x < 1$ into a number in the range $1800 \leq y < 2000$.

A2P14. Write and run a Java program that inputs the name of a month and then processes it by:

- echoing the input;
- extracting the first three letters;
- capitalizing them;
- printing that abbreviation;
- extracting each of the three letters as a separate char variable;
- using switch statement to identify the number of the month from the char variables/abbreviation;
- print the number of the month.

Here is a sample run:

```
Enter the month: February
You entered February
Its abbreviation is FEB
This is month number 2
```

A2P15. Modify the program for problem A2P14 replacing the switch statement with 12 parallel if statement. For example,

If (month.equals("FEB")) n = 2;

A2P16. Modify the program for problem of A2P15. Define the string

String months = "JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC";

Then use **months.indexOf(month)** to obtain an integer that you can use to compute the month number, there by replacing the 12 parallel if statements.

A2P17. Extend the program for problem of A2P16 so that it also prints the number of days in the month. Here is a sample run:

Enter the month: **February**

FEBRUARY is month number 2

It has 28 days.

Think of memory efficient way to store number of days in the months.

A2P18. Write and run a program that simulate a guessing game, were the computer "thinks" of a number from 1 to 200, and the user tries to guess the number. After each wrong guess, the computer gives a hint by reporting whether that guess was too high or too low. Use the conditional operator, like this:

System.out.println("No: that's too"+ (guess<x? "low.": "high."));

Give the user 5 guesses before revealing the number. Here is the sample run:

I'm thinking of a number from 1 to 100.

Guess what it is: 55

No: that's too low.

Guess again: 77

No: that's too high.

Guess again: 66

No: that's too low.

Guess again: 71

No: that's too high.

Guess again: 68

No: the number was 70

-----X-----