



Subject Code: IMT-36

Subject Name : JAVA PROGRAMMING

Notes:

- Write answers in your own words as far as possible and refrain from copying from the text books/handouts.*
- Answers of Ist Set (Part-A), IInd Set (Part-B), IIIrd Set (Part – C) and Set-IV (Case Study) must be sent together.*
- Mail the answer sheets alongwith the copy of assignments for evaluation & return.*
- Only hand written assignments shall be accepted.*

- First Set of Assignments: 5 Questions, each question carries 1.5 marks.
- Second Set of Assignments: 5 Questions, each question carries 1.5 marks.
- Third Set of Assignments: 5 Questions, each question carries 1.5 marks. Confine your answers to 150 to 200 Words.
- Forth Set of Assignments: Two Case Studies : 7.5 Marks. Each case study carries 3.75 marks.

ASSIGNMENTS

PART– A

- Define each of the following terms: single inheritance, multiple inheritance, interface, superclass and subclass.
- Discuss why casting a superclass reference to a subclass reference is potentially dangerous multiple inheritance? What feature of Java helps realize the benefits of multiple inheritance?
- Distinguish between non-abstract methods and abstract methods.
- Write a recursive method power (base, exponent) that when invoked returns base exponent
- Selection sort searches an array for the smallest element in the array. Then, the smallest element is swapped with the first element of the array. The process is repeated for the subarray beginning with the second element of the array. Each pass of the array results in one element being placed in its proper location. This sort has a performance comparably to the bubble sort-for an array n elements, n-1 passes must be made, and for each subarray, n-1 comparisons must be made to find the smallest value. When the subarray being processed contains one element, the array is sorted. Write a recursive method selection Sort to perform this algorithm.

PART– B

- Create a Data class with the following capabilities:
 - Output the date in multiple formats such as
MM/DD/YY
June 14,2009
DDD YYYY
 - Use overload constructors to create Date objects initialized with dates of the formats in part(a)
- Create class SavingAccount. Use a static class variable to store the annual interest Rate for each of the savers. 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 balance by annual interest Rate divided by 12; this interest should be

added to savingBalance. Provide a static method modifyInterestRate that sets the annual interest Rate to a new value. Write a driver program to test class SavingsAccountobject, saver1 and saver2, with balance of \$2000.00 and \$3000.00, respectively. Set annual interest Rate to 4%, then calculate the monthly interest and print the new balances for each of the savers. Then set the annual interest Rate to 5% and calculate the next month's interest and print the new balances for each of the savers.

3. Explain the advantage of object oriented model over traditional programming model.
4. Define each of the following terms: single inheritance, multiple inheritance, interface, superclass and subclass.
5. Discuss why casting a superclass reference to a subclass reference is potentially dangerous multiple inheritance? What feature of Java helps realize the benefits of multiple inheritance?

PART – C

1. Distinguish between non-abstract methods and abstract methods.
2.
 - a) What is Package.
 - b) Explain Client Server.
3.
 - a) What is LRL?
 - b) Why has Java gained international acceptance.
4.
 - a) Explain Java API.
 - b) Explain RMI with suitable example.
5.
 - a) What is package and how it is useful in programming?
 - b) Class is the heart of Every Java applet, Comment.

CASE STUDY-1

A spelling Checker: Many popular word processing software packages have built in spell checkers.

1. You are asked to develop your own spell-checker utility. We make suggestions to help get you started. You should then consider adding more capabilities. Use a computerized dictionary (if you have access to one) as a source of words.

CASE STUDY-2

Why do we type so many words with incorrect spelling? In some cases, it is because we simply do not know the correct spelling, so we make a "best guest." In some case, it is because we transpose two letters (e.g., "default" instead of "defualt"). Sometimes we double-type a letter accidentally (e.g., "hanndy" instead of "handy"). Sometimes we type a nearby key instead of the one we intended (e.g., "biryhday" instead of "birthday"). And so on.

Design and implement a spell-checker applet in Java. Your program should maintain an array wordList of strings. Enable the user to enter these strings. Once you have these capabilities, you can obtain the words for the spell checker from a computerized dictionary stored in a file.

Your program should ask a user to enter a word. The program should then look up that word in the wordList array. If the word is present in the array, your program should print "Word is spelled correctly."

If the word is not present in the array, your program should print 'word is not spelled correctly.' Then your program should try to locate other words in wordList that might be the word the user intended to type. For example, you can try all possible single transpositions of adjacent letters to discover that the word "default" is a direct match to a word in wordList. Of course, this implies that your program will check all other single transpositions such as "edfault," "dfeault," "default," "deafult," and "defualt." When you find a new word that matches one in wordList, print that word in a message such as, 'Did you mean "default?". "

2. Implement other tests such as replacing each double letter with a single letter and any other test you can develop to improve the value of your spell checker.