# ASSIGNMENT QUESTIONS

1. WAP to read values of a, b & c & display value of x where x=a/b-c.
2. WAP to convert temperature in Fahrenheit to Celsius.
3. Create a bio-data or resume on your name having student name, address, father's name, telephone no, email address, academic qualification, extracurricular activities etc.
4. Write a function to swap the values of a pair of integers using call by reference.
5. WAP to calculate the variance & standard deviation of N numbers.
6. WAP to allocate an array of n size, print the sum and average & delete the array.
7. An electricity board charges-
   For the first 100 units – 60 p per unit.
   For next 200 units- 80 p per unit.
   Beyond 300 unit- 90 p per unit.

   All users are charged a minimum of Rs 50.00. If the total amount is more than Rs 300.00 then an additional surcharge of 15% is added. WAP to read the names of users & no. of units consumed. Also, print out the charges with names.
8. WAP to calculate the simple interest and compound interest for a given principal and time period.
9. Write a function power() to raise a number **m** to a power **n.** the function takes a double value for **m** and int value for **n**, and returns the result correctly. Use a default value of 2 for n to make the function to calculate squares when this argument is omitted.

10. Write a function that performs the same operation as above but takes an int value for m. both the functions should have same name. Use the concept of function overloading.
11. WAP to enter a list of books for which we place an order with a dealer. The list includes books serial number, quantity and price of each book. Also perform operations such as adding a book to the list, delete a book from list and printing the total value of the order.
12. Define a class to represent a bank account. Include the following data members: Name of the depositor, Account number, type of the account, balance amount. Member Functions: To assign initial value, to deposit an amount, to withdraw an amount, to display name and balance. Write a program to test the class.
13. Modify the above class for handling 10 customers.
14. Define a class of employees, containing employee number, name, address and no.of dependents for the employee. Member function: to insert and display information. Define an array of 20 employees. Display all employees with more than two dependents.
15. Write a function for finding the average age of a class student. Pass an array of student objects as parameter to that function. Assume the default class strength to be 50 as default argument.
16. A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person

inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and request for the number of copies required. If the requested copies are available, the total cost of requested copies is displayed; otherwise the message “required copies not in stock” is displayed. Design a system using a class called books with suitable member functions and constructors. Use new operator in constructor to allocate memory space required.

17. Create a class MAT of size **m x n.** Define all possible matrix operations for MAT type objects.
18. Define a class string. Use overloaded == operator to compare two strings.
19. Create a class account that stores customer name, account number and type of account. From this, derive the classes cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks: accept deposit from a customer and update the balance, display the balance, compute and deposit interest, permit withdrawal and update the balance, check for the minimum balance, impose penalty, and update the balance. Do not use any constructors. Use member functions to initialize class members.
20. Create a base class called shape. Derive two specific classes called triangle and rectangle from the base shape. Using these three classes design a program that will

accept dimensions of a triangle or a rectangle interactively, and display the area.

21. Write a function template for finding the minimum value contained in an array.
22. Write a template function “alloc” that takes two parameter: **n:** size of the array to allocate, **val:** a value of type T. The alloc function should allocate an array of type T with n elements & set all the elements in the array to the value val. A pointer to array is returned.
23. Write a class template to represent a generic vector. Include member functions to do the following tasks: to create the vector, to modify the value of a given element, to multiply by a scalar value, to display the vector in the form (10, 20, 30,.....).
24. Write a program to implement the stack data structures. Include member functions push and pop with proper exception handling.
25. Write a main function that calls a deeply nested function containing an exception. Incorporate necessary exception handling mechanism.