Simple C Programs

1. Accept the radius from user and compute the area and circumference of a circle.
2. Accept a character from user and display ASCII value of it.
3. Accept marks of 5 subjects (out of 100) of a student and display total marks and compute the percentage also.
4. Accept the basic salary of an employee and compute the net salary after adding earnings and subtracting deductions.
   PF is 2 % of
   basic Tax is 3 %
   of basic HRA is
   5 % basic DA is
   8 % of basic
5. Accept two numbers and swap two numbers using
i) Third variable
ii) By performing arithmetic operations.

6. Accept dimensions of a cylinder and print the surface area and volume (Hint: surface area = 2πr 2 +2πrh, volume = π r 2 h). Define a constant variable pi=3.14.
7. Accept temperatures in Fahrenheit (F) and print it in Celsius(C) and Kelvin (K) (Hint: C=5/9(F-32), K = C + 273.15)

If - else

1. Write a program to accept an integer and check if it is even or odd.
2. Write a program to accept a number and check if it is divisible by 5 and 7.
3. Write a program, which accepts annual basic salary of an employee and calculates and displays the Income tax as per the following rules.
Basic: < 1, 50,000 Tax = 0
   1, 50,000 to 3,00,000 Tax = 20%
   > 3,00,000 Tax = 30%
4. Accept a lowercase character from the user and check whether the character is a vowel or consonant. (Hint: a, e, i, o, u are vowels)
5. Write a C program to input angles of a triangle and check whether triangle is valid or not.

6. Write a C program to check whether a entered character is uppercase or lowercase alphabet.

7. Write a C program to accept a character and invert the case of it.

8. Write a program to accept 3 numbers and compute minimum and maximum from them.

Switch - case

1. Accept a single digit from the user and display it in words. For example, if digit entered is 9, display Nine.
2. Write a program, which accepts two integers and an operator as a character (+ - * /), performs the corresponding operation and displays the result.
3. Accept two numbers in variables x and y from the user and perform the following operations

Options	Actions
1.	Equality Check if x is equal to y
2.	Less Than Check if x is less than y
3.	Quotient and Remainder Divide x by y and display the quotient and remainder
4.	Range : Accept a number and check if it lies between x and y (both

	inclusive)
5.	Swap : Interchange x and y

4. Accept radius from the user and write a program having menu with the following options and corresponding actions:

Options	**Actions**
1.	Area of Circle
2.	Circumference of Circle
3.	Volume of Sphere

5. Write a program having menu that has three options - add, subtract or multiply two fractions. The two fractions and the options are taken as input and the result is displayed as output. Each fraction is read as two integers, numerator and denominator.

Loops

1. Write a program that accepts numbers continuously as long as the number is positive and prints the sum of the given numbers.
2. Write a program to accept two integers x and n and compute x raised to n.
3. Write a program to accept a character, an integer n and display the next n characters.
4. Write a program to calculate factorial of a number. For e.g. factorial of 5 = 5! = 5 *4*3*2*1 = 120
5. Write a program to calculate factors of a given number.
6. Accept two numbers and calculate GCD of them.
7. Write a menu driven program to do following operations :
   a) Compute area of circle
   b) Compute area of rectangle
   c) Compute area of triangle
   d) Exit
   Display menu, ask choice to the user, depending on choice accept the parameters and perform the operation. Continue this process until user selects exit option.
8. Write a program to print all prime numbers between 1 to n.
9. Write a program to solve following pyramid pattern

```
a)*          | b) 1          |c)   A
  * *        |    1 2        |     A B
  * * *      |    1 2 3      |     A B C
  * * * *    |    1 2 3 4    |     A B C D
  * * * * *  |    1 2 3 4 5  |     A B C D E
```

10. Write a program to solve following pyramid pattern

a)
```
* * * * *
  * * * *
    * * *
      * *
        *
```

b)
```
* * * * * * * * *
  * * * * * * *
    * * * * *
      * * *
        *
```

Array

1D array

1. Write a program to accept n numbers in an array and display the largest and smallest number. Using these values, calculate the range of elements in the array.
2. Write a program to accept an array of n elements and a number say key. Check whether key is present in the array or not.
3. Write a program to accept an integer array and an integer say num and counts the occurrences of the num in the array.
4. Write a program to accept n numbers from the user and store them in an array. Then sort the array in descending order and display it.
5. Write a program to accept a decimal number and convert it to binary.
6. Program to print all even numbers in array.

2D array

1. Write a program to accept, display and print the sum of elements of each row and sum of elements of each column of a matrix.
2. Write a program to accept a matrix A of size mXn and store its transpose in matrix B. Display matrix B.
3. Write a program to add and multiply two matrices. Perform necessary checks before adding and multiplying the matrices.

4. Write a program to perform the following operations on a square matrix. Write
   i) Check if the matrix is symmetric.
   ii) Display the trace of the matrix (sum of diagonal elements).
   iii) Check if the matrix is an upper triangular matrix.

String

1. Write a program which accepts a sentence from the user and alters it as follows: Every space is replaced by *, case of all alphabets is reversed, digits are replaced by?

2. Write a program that accepts n strings and displays the longest string. Use array of strings.

3. Write a menu driven program to perform the following operations on strings using standard library functions: Length, Copy, Concatenation, Compare, Reverse, Uppercase, Lowercase, Check case.

4. Write a program to count Occurrence Of Vowels & Consonants in a String.

5. Create user defined strrev(),strcat(),strcpy(),strlen() with help of array notation & pointer notation

Pointer and Function

1. WAP to accept an array of n integers and calculate sum of odd numbers and even numbers using the pointer to an array.
2. Write a function isEven, which accepts an integer as parameter and returns 1 if the number is even, and 0 otherwise. Use this function in main to accept n numbers and ckeck if they are even or odd.
3. Write a function isPrime, which accepts an integer as parameter and returns 1 if the number is prime and 0 otherwise. Use this function in main to display the first 10 prime numbers.
4. For the following standard functions, write corresponding user defined functions and write a menu driven program to use them. strcat, strcmp, strrev, strupr.
5. Write a function power, which calculates x raised to y . Write another function, which calculates n! Using for loop. Use these functions to calculate the sum of first n terms of the Taylor series:
   sin(x) = x - 3! 3 x + 5! 5 x + ……
6. Write a recursive C function to calculate the GCD of two numbers. Use this function in main. The GCD is calculated as :
   gcd(a,b) = a if b = 0
   = gcd (b, a mod b) otherwise.
7. Write a recursive C function to calculate x raised to y . (Do not use standard library function)
8. Write a recursive function to calculate the sum of digits of a number till you get a single digit number. Example: 961 -> 16 -> 5. (Note: Do not use a loop)
9. Write a recursive function to calculate the nth Fibonacci number. Use this function in main to display the first n Fibonacci numbers.
   The recursive definition of nth Fibonacci number is as follows: fib(n) = 0 if n = 1
   = 1 if n = 2
   = fib(n-2) + fib(n-1) if n>2

Structure

1. Create a structure student (roll number, name, marks of 3 subjects, percentage). Accept details of n students and write a menu driven program to perform the following operations. Write separate functions for the different options.
   i) Search
   ii) Modify
   iii) Display all student details
   iv) Display all student having percentage > 80
   v) Display student having maximum percentage

2. Create a structure employee (id, name, salary). Accept details of n employees and write a menu driven program to perform the following operations. Write separate functions for the different options.
   i) Search by name
   ii) Search by id
   iii) Display all
   iv) Display all employees having salary > 25000
   v) Display employee having maximum

3. The following structure is for a library book with the following details: id, title, publisher, code (1 – Text book, 2 – Magazine, 3 – Reference book). If the code is 1, store no-of-copies. If code = 2, store the issue month name. If code = 3, store edition number. Also store the cost.

```
struct library_book
{
   int id;
   char title[80];
   char publisher[20]
   ; int code;
    union u {
        int
        no_of_copies;
        char month[10];
        int edition;
   }info;
  int
  cost;
};
```

Write a program to accept details of n books. Use switch - case to accept the code and details according to the code.

****

1. Write a program to create student class with data members rollno, marks1, mark2, mark3. Accept data (acceptInfo()) and display using display member function.
Also display total, percentage and grade.

2. Create a class Person with data members as name, age, city. Write getters and setters for all the data members. Also add the display function. Create Default and Parameterized constructors. Create the object of this class in main method and invoke all the methods in that class.

3. Create a class Date with data members as dd, mm, yy. Write getters and setters for all the data members. Also add the display function. Create Default and Parameterized constructors. Create the object of this class in main method and invoke all the methods in that class.
4. Create a class Book with data members as bname, id, author, price. Write getters and setters for all the data members. Also add the display function. Create Default and Parameterized constructors. Create the object of this class in main method and invoke all the methods in that class.
5. Create a class Point with data members as x,y. Create Default and Parameterized constructors. Write getters and setters for all the data members. Also add the display function. Create the object of this class in main method and invoke all the methods in that class.
6. Create a class ComplexNumber with data members real, imaginary. Create Default and Parameterized constructors. Write getters and setters for all the data members. Also add the display function. Create the object of this class in main method and invoke all the methods in that class.
7. Create Date class with members day, month, year. Write no argument and parameterized constructor. Create two object s and initialize them using no argument and parameterized constructor respectively. Print date using display function.
8. Create Employee class with members id(int),name(string),dob(Date).Use above created Date class. Write default and parameterized constructor in Employee Class.Write accept () function to accept information and display () to display emp information.
9. Consider that payroll software needs to be developed for computerization of operations of an ABC organization. The organization has employees.

9.1. Construct a class Employee with following members using private access specifies: Employee Id integer
Employee Name
string Basic Salary
double HRA double
Medical
double=1000 PF
double
PT double
Net Salary double
Gross Salary
double

Please use following expressions for calculations: //Note: Don't accept HRA,PF PT from user
* HRA = 50% of Basic Salary
* PF = 12% of Basic Salary
* PT = Rs. 200

9.2. Write methods to display the details of an employee and calculate the gross and net salary.
* Goss Salary = Basic Salary + HRA + Medical
* Net Salary = Gross Salary – (PT + PF)

9.3 Create Object of employee class and assign values and display Details.

10. Solve this.

Fresh business scenario to apply inheritance , polymorphism to emp based organization scenario. Create Emp based organization structure --- Emp , Mgr , Worker

10.1 Emp state--- id(int), name, deptId ,
basicSalary(double) Accept all of above in constructor

arguments.
Methods ---
10.2. Compute net salary ---ret 0

(eg : public double computeNetSalary(){return 0;})

10.3 Mgr state ---id,name,basic,deptId ,
perfBonus Add suitable constructor
Methods ----
1. Compute net salary (formula: basic+perfBonus) -- override computeNetSalary

10.4 Worker state --
id,name,basic,deptId,hoursWorked,hourlyRate Methods:
1. Compute net salary (formula: = basic+ (hoursWorked*hourlyRate) --override computeNetSalary
2. Get hrlyRate of the worker -- add a new method to return hourly rate of a worker.(getter)

Create suitable array to store organization
details. Provide following options
1. Hire Manager
I/P : all manager details

2. Hire Worker
I/P : all worker details

3. Display information of all employees net salary (by invoking computeNetSal),

4. Exit
---

11. Create cpp application for bank account handling.
11.1. Create a class BankAccount -- acct no(int),customer
name(string),balance(double) Add constr. (2 constrs : first to accept all details)
11.2 Add Business logic
methods Methods
public void withdraw(double
amt) public void deposit(double
amt)

11.3 : Create object of account class and test withdraw and deposit methods.
---

12. Create an abstract class Shape with pure virtual method area;
Create Rectangle, Circle, Square class. Inherit them from Shape class. Override area method. Test these all classes by creating object of respective class.