**OBJECT ORIENTED PROGRAMMING**

**C++ - DSA01**

**OPERATOR AND CONTROL STRUCTURES**

1. Write a program to read in two integers and perform the following operations on them: addition, subtraction, multiplication, division, and modulo.
2. Program to determine the integer is odd or even
3. Program to compute the average of three integers
4. Program to check two numbers are equal or not
5. Write a program to read in two Floating numbers and perform the following operations on them: addition, subtraction, multiplication, division, and modulo.
6. Program to check the character is a vowel or consonant
7. Program to check the number is positive, negative or zero
8. Program to determine which number is greater among two integers
9. Program to read a floating-number and round it to the nearest integer using the floor an ceil functions.
10. Program to swap two numbers using bitwise XOR operator
11. Largest among three numbers using ternary conditional operator
12. Program to check two numbers are equal or not using ternary conditional operator
13. Program to check the integer is divisible by 3 or not using ternary conditional operator
14. Program to print numbers from 1 to 10 using for loop
15. Factorial of a number using for loop
16. Print multiplication table using for loop
17. Fibonacci series using for loop
18. Prime number using for loop
19. Check the string is palindrome or not using while loop
20. Sum of all digits using while loop (n=123 output:1+2+3=6)
21. GCD of two numbers using do-while loop
22. Check whether the number is perfect or not
23. Armstrong number
24. Harshad number
25. Happy number
26. strong number
27. buzz number
28. neon number
29. abundant number
30. narcissistic number
31. print the pattern 1 22 333 4444 55555
32. print the pattern \* \*\* \*\*\* \*\*\*\* \*\*\*\*\*
33. Print pascal triangle pattern nested for loop
34. Print diamond pattern with \* using nested for loop
35. Program to reverse the elements in an array
36. Program to insert an element in an array at a specific position
37. Program to Delete an element in an array at a specific position
38. Find the sum of all elements in an array
39. Find the average of all elements in an array
40. Find the second largest element in an array
41. Find the number of occurrences of a value in an array
42. Merge two array
43. Create a dynamic array using pointers and display the values
44. Create a dynamic 2D (Two dimensional) array using pointers and display the values
45. Add 2 matrices
46. Multiply 2 matrices
47. Find the sum of diagonals of a matrix

**Functions in C++**

1. Find factorial using function
2. Find prime number using function
3. Find the reverse of a string using function
4. Find minimum and maximum element in an array using function
5. Find GCD of two number using function
6. Function to count the no of elements in a string
7. Convert Celsius and Fahrenheit using function
8. Find the area of a circle using function
9. Check whether the string is palindrome or not

**Constructor and destructor**

1. Write a c++ program to create a class for a bank account with a constructor and a destructor
2. Write a c++ program to create a class for a car with a constructor and a destructor
3. Write a c++ program to create a class for a rectangle with a constructor and a destructor
4. Write a c++ program to create a class for a book with a constructor and a destructor
5. Write a c++ program to create a class for student with a constructor and a destructor

**Operator overloading**

1. Write a c++ program to overload the ++ operator to increment a variable
2. Write a c++ program to overload the + operator to add two variables
3. Write a c++ program to overload the << operator to print contents of a user defined class
4. Write a c++ program to overload the == operator to compare two objects of a user defined class
5. Write a c++ program to overload the \* operator to multiply two matrices
6. rite a c++ program to overload the [] operator to access the elements in an array using index values
7. Write a c++ program to overload the () to call a function with arguments
8. rite a c++ program to overload the – operator to subtract two variables
9. write a c++ program to overload a function to add two integer numbers and two floating point number separately
10. Write a c++ program to overload the += operator to add two objects of a user defined class
11. write a c++ program to overload a function to find the maximum value from two integer numbers and two floating point number, and two characters separately
12. write a c++ program to overload a function to concatenate two strings and two characters arrays separately
13. write a c++ program to overload a function to calculate the sum of two matrices and two arrays separately
14. write a c++ program to overload a function to print an integer array, a double array and a character array separately
15. write a c++ program to overload a function to find a factorial of an integer number and factorial of a floating-point number separately
16. write a c++ program to overload a function to sort an integer array and a double array
17. write a c++ program to overload a function to calculate the power of an integer number and power of a floating-point number separately
18. write a c++ program to overload a function to find an absolute value of an integer number and absolute value of a floating-point number separately

**Inheritance and pointers**

1. Create a base class called Shape with data members for height and width. Derive two classes Rectangle and Triangle from the base class. Write member functions to calculate the area and perimeter of each class
2. Create a base class called vehicle with data members for make, model, and year. Derive two classes Car and Truck from the base class. The Car class should have additional data members for seating capacity and fuel type, while the Truck class should have additional data members for payload capacity and towing capacity. Write member functions to get and set the data members for each class
3. Create a base class called Animal with data members for name, species, and age. Derive two classes Cat and Dog from the base class. The Cat class should have additional data members for color and breed, while the Dog class should have additional data members for weight and breed. Write member functions to get and set the data members for each class
4. Create a base class called Employee with data members for name, d, and salary Derive two classes Manager and Engineer from the base class. The Manager class should have additional data members for department and bonus, while the Engineer class should have additional data members for specialty and hours. Write member functions to get and set the data members for each class
5. Create a base class called Person with data members for name, age, and gender. Derive two classes Student and Teacher from the base class. The Student class should have additional data members for roll number and class, while the Teacher class should have additional data members for subject and salary. Write member functions to get and set the data members for each class.
6. Write a C++ program to create a pointer to an integer and display its value.
7. Write a C++ program to create a pointer to a float and display its value.
8. Write a C++ program to create a pointer to a char and display its value.
9. Write a C++ program to create a pointer to a double and display its value.
10. Write a C++ program to create a pointer to a string and display its value.
11. Write a C++ program to create a pointer to an array of elements and display its value.
12. Write a C++ program to create a pointer to an array of character and display its value.
13. Write a C++ program to create a pointer to an array of floats and display its value.
14. Write a C++ program to create a pointer to an object and display its attributes.
15. Write a C++ program to create a pointer to a function and call the function using the pointer.

**Polymorphism**

1. Create a base class called Person with a virtual function work (). Derive two classes Employee and Manager from the base class. Implement the work () function for each class
2. Create a base class called Animal with a virtual function eat (). Derive two classes Herbivore and Carnivore from the base class. Implement the eat function for each class.
3. Create a base class called Shape with virtual functions area () and volume (). Derive two classes Sphere and Cylinder from the base class. Implement the area and volume () functions for each class
4. Create a base class called Person with a virtual function greet). Derive two classes Student and Teacher from the base class. implement the greet) function for each class
5. Create a base class called Person with a virtual function greet). Derive two classes Student and Teacher from the base class. implement the greet) function for each class
6. Create a base class called Shape with virtual functions area( ) and perimeter(). Derive two classes Rectangle and Triangle from the base class. Implement the area () and perimeter () functions for each class.
7. Create a base class called Vehicle with a virtual function drive(). Derive two classes Car and Truck from the base class. Implement the drive() function for each class.
8. Create a base class called Employee with a virtual function calculate Pay(). Derive two classes Manager and Engineer from the base class. Implement the calculatePay () function for each class.
9. Create a base class called Animal with a virtual function speak(). Derive two classes Cat and Dog from the base class. Implement the speak() function for each class.
10. Create a base class called Shape with a virtual function area(). Derive two classes Rectangle and Circle from the base class. Implement the area() function for each class.

**Exception Handling**

1. Write a c++ program to demonstrate to use of the finally block for handling exceptions
2. Write a c++ program to demonstrate to use of nested try-catch blocks for handling exceptions
3. Write a c++ program to demonstrate to use of user-defined exception for handling custom exception
4. Write a c++ program to demonstrate to use of the standard class for handling exceptions
5. Write a c++ program to demonstrate to use of the keyword to throw an exception
6. Write a c++ program to demonstrate to use of multiple catch blocks for handling different types of exceptions
7. Write a c++ program to demonstrate to use of try-catch blocks for handling exceptions