Set1

Q1A: Write a program to input two numbers and display their sum.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

// Declare variables to store input numbers

float num1, num2, sum;

// Input two numbers from the user

cout << "Enter the first number: ";

cin >> num1;

cout << "Enter the second number: ";

cin >> num2;

// Calculate the sum of the two numbers

sum = num1 + num2;

// Display the sum

cout << "The sum of " << num1 << " and " << num2 << " is: " << sum << endl;

return 0;

}

Q2Write a program to check if a given year is a leap year.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int year;

// Input year from user

cout << "Enter a year: ";

cin >> year;

// Check if the year is a leap year

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

cout << year << " is a leap year." << endl;

} else {

cout << year << " is not a leap year." << endl;

}

return 0;

}

Q3: Write a program to demonstrate simple inheritance by creating a base class Animal with an attribute name and a derived class Dog that adds a method bark(). (6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

// Base class Animal

class Animal {

public:

// Attribute for the animal's name

string name;

// Constructor to initialize the name

Animal(string n) : name(n) {}

// Method to display the name of the animal

void display() {

cout << "Animal Name: " << name << endl;

}

};

// Derived class Dog inheriting from Animal

class Dog : public Animal {

public:

// Constructor to initialize the name (calls base class constructor)

Dog(string n) : Animal(n) {}

// Method to make the dog bark

void bark() {

cout << name << " says: Woof! Woof!" << endl;

}

};

int main() {

// Create an object of the Dog class

Dog myDog("Buddy");

// Call the display method of Animal class

myDog.display();

// Call the bark method of Dog class

myDog.bark();

return 0;

}

Set2

Q1Write a program to calculate the area of a rectangle by taking its length and width as input. (6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

// Declare variables for length, width, and area

float length, width, area;

// Input the length of the rectangle

cout << "Enter the length of the rectangle: ";

cin >> length;

// Input the width of the rectangle

cout << "Enter the width of the rectangle: ";

cin >> width;

// Calculate the area of the rectangle

area = length \* width;

// Output the area of the rectangle

cout << "The area of the rectangle is: " << area << endl;

return 0;

}

Q2: Write a program to check if a person is eligible to vote (age 18 or above).(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

// Declare a variable to store the person's age

int age;

// Prompt the user to enter their age

cout << "Enter your age: ";

cin >> age;

// Check if the person is eligible to vote

if (age >= 18) {

cout << "You are eligible to vote." << endl;

} else {

cout << "You are not eligible to vote." << endl;

}

return 0;

}

Q3: Write a program to input a positive integer N and calculate the sum of the first N natural numbers using a for loop.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

// Declare a variable to store the number N and the sum

int N, sum = 0;

// Input the value of N

cout << "Enter a positive integer N: ";

cin >> N;

// Check if the entered number is positive

if (N <= 0) {

cout << "Please enter a positive integer." << endl;

return 1; // Exit the program if the input is not positive

}

// Use a for loop to calculate the sum of the first N natural numbers

for (int i = 1; i <= N; i++) {

sum += i; // Add the current number to sum

}

// Output the sum

cout << "The sum of the first " << N << " natural numbers is: " << sum << endl;

return 0;

}

Set3

Q1: Write a program to find the product of three numbers entered by the user.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

// Declare variables to store three numbers

float num1, num2, num3, product;

// Input the three numbers from the user

cout << "Enter the first number: ";

cin >> num1;

cout << "Enter the second number: ";

cin >> num2;

cout << "Enter the third number: ";

cin >> num3;

// Calculate the product of the three numbers

product = num1 \* num2 \* num3;

// Output the result

cout << "The product of " << num1 << ", " << num2 << " and " << num3 << " is: " << product << endl;

return 0;

}

Q2Write a program to check if a password is valid. A password is valid if it has at least 8 characters and does not contain any spaces.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

#include <string>

using namespace std;

int main() {

// Declare a variable to store the password

string password;

// Input the password from the user

cout << "Enter your password: ";

getline(cin, password); // Use getline to read spaces

// Check if the password has at least 8 characters and contains no spaces

if (password.length() >= 8 && password.find(' ') == string::npos) {

cout << "Password is valid." << endl;

} else {

cout << "Password is invalid." << endl;

}

return 0;

}

Q3: Write a program to check if a number entered by the user is positive, negative, or zero.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

// Declare a variable to store the number

int number;

// Input the number from the user

cout << "Enter a number: ";

cin >> number;

// Check if the number is positive, negative, or zero

if (number > 0) {

cout << "The number is positive." << endl;

} else if (number < 0) {

cout << "The number is negative." << endl;

} else {

cout << "The number is zero." << endl;

}

return 0;

}

Set4

Q1: Write a program to take two boolean inputs (1 for true, 0 for false) and display the results of AND (&&) and OR (||) operations.(6Marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

// Declare two boolean variables to store the inputs

bool a, b;

// Input two boolean values (1 for true, 0 for false)

cout << "Enter the first boolean value (1 for true, 0 for false): ";

cin >> a;

cout << "Enter the second boolean value (1 for true, 0 for false): ";

cin >> b;

// Perform AND (&&) operation

bool andResult = a && b;

cout << "The result of AND (a && b) is: " << andResult << endl;

// Perform OR (||) operation

bool orResult = a || b;

cout << "The result of OR (a || b) is: " << orResult << endl;

return 0;

}

Q2Write a program to assign a grade based on marks: •90 and above: A •80 to 89 : B •70 to 79 : C •Below 70 : Fail. (6Marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

// Declare a variable to store marks

int marks;

// Input marks from the user

cout << "Enter the marks: ";

cin >> marks;

// Check the range of marks and assign a grade

if (marks >= 90) {

cout << "Grade: A" << endl;

} else if (marks >= 80) {

cout << "Grade: B" << endl;

} else if (marks >= 70) {

cout << "Grade: C" << endl;

} else {

cout << "Grade: Fail" << endl;

}

return 0;

}

Q3: Write a program to calculate the factorial of a number using recursion.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

// Recursive function to calculate factorial

int factorial(int n) {

if (n <= 1) {

return 1; // Base case: factorial of 0 or 1 is 1

} else {

return n \* factorial(n - 1); // Recursive call

}

}

int main() {

// Declare a variable to store the number

int num;

// Input the number from the user

cout << "Enter a number: ";

cin >> num;

// Check for non-negative input

if (num < 0) {

cout << "Factorial is not defined for negative numbers." << endl;

} else {

// Call the recursive function and display the result

cout << "The factorial of " << num << " is: " << factorial(num) << endl;

}

return 0;

}

Set5

Q1Write a program to check if a number is divisible by 5 and 3..(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

// Declare a variable to store the number

int number;

// Input the number from the user

cout << "Enter a number: ";

cin >> number;

// Check if the number is divisible by both 5 and 3

if (number % 5 == 0 && number % 3 == 0) {

cout << "The number is divisible by both 5 and 3." << endl;

} else {

cout << "The number is NOT divisible by both 5 and 3." << endl;

}

return 0;

Q2: Write a program to input a number (1 to 12) and display the corresponding month name using a switch statement (6Marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

// Declare a variable to store the month number

int month;

// Input the month number from the user

cout << "Enter a number (1 to 12): ";

cin >> month;

// Use switch statement to display the corresponding month name

switch(month) {

case 1:

cout << "January" << endl;

break;

case 2:

cout << "February" << endl;

break;

case 3:

cout << "March" << endl;

break;

case 4:

cout << "April" << endl;

break;

case 5:

cout << "May" << endl;

break;

case 6:

cout << "June" << endl;

break;

case 7:

cout << "July" << endl;

break;

case 8:

cout << "August" << endl;

break;

case 9:

cout << "September" << endl;

break;

case 10:

cout << "October" << endl;

break;

case 11:

cout << "November" << endl;

break;

case 12:

cout << "December" << endl;

break;

default:

cout << "Invalid input! Please enter a number between 1 and 12." << endl;

break;

}

return 0;

}

Q3Write a program to find the largest number among three numbers entered by the user.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

// Declare variables to store three numbers

int num1, num2, num3;

// Input the three numbers from the user

cout << "Enter the first number: ";

cin >> num1;

cout << "Enter the second number: ";

cin >> num2;

cout << "Enter the third number: ";

cin >> num3;

// Compare the three numbers to find the largest

if (num1 >= num2 && num1 >= num3) {

cout << "The largest number is: " << num1 << endl;

} else if (num2 >= num1 && num2 >= num3) {

cout << "The largest number is: " << num2 << endl;

} else {

cout << "The largest number is: " << num3 << endl;

}

return 0;

}

Set- 6

1. : Write a program to declare variables for your name, age, and country, and print them.

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard namespace, so we don't need to prefix with 'std::'

x

int main() {

string name = "John Doe"; // Declare a string variable 'name' and initialize it with "John Doe"

int age = 25; // Declare an integer variable 'age' and initialize it with the value 25

string country = "USA"; // Declare a string variable 'country' and initialize it with "USA"

// Print the name, age, and country

cout << "Name: " << name << endl; // Print the value of the 'name' variable

cout << "Age: " << age << endl; // Print the value of the 'age' variable

cout << "Country: " << country << endl; // Print the value of the 'country' variable

return 0; // End of the program, return 0 to indicate successful execution

}

1. :Write a program to convert a temperature from Celsius to Fahrenheit using the formula:

F = \frac{9}{5}C + 32

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

double celsius, fahrenheit; // Declare variables to store Celsius and Fahrenheit values

// Ask the user to input a temperature in Celsius

cout << "Enter the temperature in Celsius: ";

cin >> celsius; // Read the Celsius temperature from the user

// Convert Celsius to Fahrenheit using the formula: F = (9/5) \* C + 32

fahrenheit = (9.0 / 5.0) \* celsius + 32; // Perform the conversion and store the result in 'fahrenheit'

// Display the converted temperature in Fahrenheit

cout << "The temperature in Fahrenheit is: " << fahrenheit << endl;

return 0; // End of the program

}

1. Write a program to input three numbers, calculate their average, and display the Result

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

double num1, num2, num3, average; // Declare variables to store the three numbers and their average

// Ask the user to input three numbers

cout << "Enter the first number: ";

cin >> num1; // Read the first number from the user

cout << "Enter the second number: ";

cin >> num2; // Read the second number from the user

cout << "Enter the third number: ";

cin >> num3; // Read the third number from the user

// Calculate the average of the three numbers

average = (num1 + num2 + num3) / 3; // Add the three numbers and divide by 3 to get the average

// Display the average

cout << "The average of the three numbers is: " << average << endl;

return 0; // End of the program

}

Set – 7

1. Write a program to input a string and extract a substring from it (starting from a

specified position for a specified length).

#include <iostream> // Include the input-output stream library to handle input and output

#include <string> // Include the string library to use the string class

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

string str, substring; // Declare a string variable 'str' to store the input string, and 'substring' to store the extracted substring

int startPos, length; // Declare variables for the starting position and length of the substring

// Ask the user to input a string

cout << "Enter a string: ";

getline(cin, str); // Read the full line of text (including spaces) and store it in 'str'

// Ask the user for the starting position

cout << "Enter the starting position (0-based index): ";

cin >> startPos; // Read the starting position entered by the user

// Ask the user for the length of the substring

cout << "Enter the length of the substring: ";

cin >> length; // Read the length of the substring entered by the user

// Check if the starting position and length are within valid range

if (startPos >= 0 && startPos < str.length() && length > 0 && (startPos + length) <= str.length()) {

// Extract the substring from the input string starting from 'startPos' and with the specified 'length'

substring = str.substr(startPos, length); // 'substr' extracts a substring starting at 'startPos' for 'length' characters

// Display the extracted substring

cout << "Extracted substring: " << substring << endl;

} else {

// If the user input is invalid (starting position or length is out of range), show an error message

cout << "Invalid starting position or length!" << endl;

}

return 0; // End of the program

}

1. : Write a program to input a number and find its absolute value using the abs() function.

#include <iostream> // Include the input-output stream library to handle input and output

#include <cstdlib> // Include the cstdlib library to use the abs() function

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

int num; // Declare an integer variable 'num' to store the input number

// Ask the user to input a number

cout << "Enter a number: ";

cin >> num; // Read the number entered by the user and store it in 'num'

// Find and display the absolute value of the number using the abs() function

cout << "The absolute value of " << num << " is: " << abs(num) << endl;

return 0; // End of the program

}

1. :Write a program to read the content of a text file and display it on the screen

#include <iostream> // Include the input-output stream library to handle input and output

#include <fstream> // Include the fstream library to handle file operations

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

string filename; // Declare a string variable to store the name of the file

ifstream file; // Declare an input file stream object to read from the file

// Ask the user to enter the name of the file to open

cout << "Enter the name of the file to read: ";

cin >> filename; // Read the filename from the user and store it in the 'filename' variable

// Open the file in read mode

file.open(filename); // Attempt to open the file with the name stored in 'filename'

// Check if the file opened successfully

if (!file) { // If the file could not be opened (for example, if the file does not exist)

cout << "Error opening the file!" << endl; // Display an error message

return 1; // Exit the program with an error code

}

string line; // Declare a string variable to store each line read from the file

// Read the file line by line and display its contents

while (getline(file, line)) { // Read a line from the file and store it in 'line'

cout << line << endl; // Print the line to the screen

}

file.close(); // Close the file after reading is complete

return 0; // End of the program

}

Set -8

1. Write a program to input a number and print whether it is even or odd.

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

int num; // Declare an integer variable 'num' to store the user's input number

// Ask the user to input a number

cout << "Enter a number: ";

cin >> num; // Read the number entered by the user and store it in 'num'

// Check if the number is even or odd using the modulus operator (%)

if (num % 2 == 0) { // If the remainder when num is divided by 2 is 0, the number is even

cout << num << " is even." << endl; // Print that the number is even

} else { // If the remainder is not 0, the number is odd

cout << num << " is odd." << endl; // Print that the number is odd

}

return 0; // End of the program

}

1. Write a program to input two numbers and display the result of their division. Handle any division by zero error

#include <iostream> // Include the input-output stream library for input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

double num1, num2; // Declare two double variables to store the numbers entered by the user

// Ask the user to input the first number

cout << "Enter the first number: ";

cin >> num1; // Read the first number entered by the user and store it in 'num1'

// Ask the user to input the second number

cout << "Enter the second number: ";

cin >> num2; // Read the second number entered by the user and store it in 'num2'

// Check if the second number is zero before performing division

if (num2 == 0) { // If num2 is zero, division by zero would occur

cout << "Error: Division by zero is not allowed." << endl; // Display an error message

} else { // If num2 is not zero, perform the division

double result = num1 / num2; // Perform the division and store the result in 'result'

cout << "The result of " << num1 << " divided by " << num2 << " is: " << result << endl; // Display the result

}

return 0; // End of the program

}

1. Write a program to read from a file. If the file doesn’t exist, use exception handling to display an error message

#include <iostream> // Include the input-output stream library for handling input and output

#include <fstream> // Include the fstream library for file input/output operations

#include <stdexcept> // Include the standard library for exception handling

using namespace std; // Use the standard namespace to avoid needing to prefix with 'std::'

int main() {

string filename; // Declare a string variable to store the name of the file

ifstream file; // Declare an ifstream object to read from the file

// Ask the user to input the filename

cout << "Enter the name of the file to read: ";

cin >> filename; // Read the file name entered by the user and store it in 'filename'

try {

file.open(filename); // Attempt to open the file in read mode

// Check if the file is open and accessible

if (!file) { // If the file couldn't be opened (e.g., it doesn't exist)

throw runtime\_error("Error: The file could not be opened!"); // Throw an exception with an error message

}

string line; // Declare a string to store each line read from the file

// Read the file line by line

while (getline(file, line)) { // Read each line from the file

cout << line << endl; // Print each line to the console

}

file.close(); // Close the file after reading is done

} catch (const runtime\_error& e) { // Catch runtime\_error exceptions

cout << e.what() << endl; // Print the error message from the exception

}

return 0; // Return 0 to indicate the program ended successfully

}

Set – 9

1. Write a program to define a structure Student with attributes name, rollNo, and marks. Create an instance of the structure and display the values

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

struct Student { // Define a structure named 'Student'

string name; // Declare a string variable 'name' to store the student's name

int rollNo; // Declare an integer variable 'rollNo' to store the student's roll number

float marks; // Declare a float variable 'marks' to store the student's marks

};

int main() {

// Create an instance of the Student structure

Student student1;

// Assign values to the attributes of the student instance

student1.name = "John Doe"; // Assign the name "John Doe" to the 'name' attribute

student1.rollNo = 101; // Assign the roll number 101 to the 'rollNo' attribute

student1.marks = 85.5; // Assign the marks 85.5 to the 'marks' attribute

// Display the values of the student's attributes

cout << "Student Name: " << student1.name << endl; // Print the student's name

cout << "Roll Number: " << student1.rollNo << endl; // Print the student's roll number

cout << "Marks: " << student1.marks << endl; // Print the student's marks

return 0; // End of the program

}

1. Write a program to find the length of a string

#include <iostream> // Include the input-output stream library for input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

string str; // Declare a string variable to store the input string

// Ask the user to input a string

cout << "Enter a string: ";

getline(cin, str); // Read the entire line entered by the user and store it in 'str'

// Find the length of the string using the built-in length() function

int length = str.length(); // 'length()' function returns the number of characters in the string

// Display the length of the string

cout << "The length of the string is: " << length << endl;

return 0; // Return 0 to indicate successful execution of the program

}

1. : Write a program to convert a temperature from Celsius to Fahrenheit using the formula:

F = \frac{9}{5}C + 32

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard namespace (no need to prefix 'std::')

int main() {

double celsius, fahrenheit; // Declare variables for Celsius and Fahrenheit temperatures

// Ask the user to enter a temperature in Celsius

cout << "Enter the temperature in Celsius: ";

cin >> celsius; // Read the temperature entered by the user and store it in 'celsius'

// Convert the Celsius temperature to Fahrenheit using the formula F = (9/5) \* C + 32

fahrenheit = (9.0 / 5.0) \* celsius + 32; // Perform the conversion and store the result in 'fahrenheit'

// Display the temperature in Fahrenheit

cout << "The temperature in Fahrenheit is: " << fahrenheit << endl;

return 0; // Return 0 to indicate successful execution of the program

}

Set – 10

1. Write a program to find the minimum of two numbers entered by the user

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

int num1, num2; // Declare two integer variables to store the numbers entered by the user

// Ask the user to input two numbers

cout << "Enter the first number: ";

cin >> num1; // Read the first number and store it in 'num1'

cout << "Enter the second number: ";

cin >> num2; // Read the second number and store it in 'num2'

// Compare the two numbers and find the minimum

if (num1 < num2) { // If the first number is smaller than the second

cout << "The minimum number is: " << num1 << endl; // Print the first number as the minimum

} else { // Otherwise, the second number is smaller or equal

cout << "The minimum number is: " << num2 << endl; // Print the second number as the minimum

}

return 0; // Return 0 to indicate successful execution of the program

}

1. Write a program to find the square and cube of a number entered by the user

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard namespace (no need to prefix 'std::')

int main() {

double num; // Declare a variable to store the number entered by the user

// Ask the user to input a number

cout << "Enter a number: ";

cin >> num; // Read the number entered by the user and store it in 'num'

// Calculate the square of the number

double square = num \* num; // Square the number by multiplying it by itself

// Calculate the cube of the number

double cube = num \* num \* num; // Cube the number by multiplying it by itself twice

// Display the square and cube of the number

cout << "The square of " << num << " is: " << square << endl;

cout << "The cube of " << num << " is: " << cube << endl;

return 0; // Return 0 to indicate successful execution of the program

}

:3) Write a program to input a number and use the += operator to add 10 to the number, then display the updated value

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard namespace (no need to prefix 'std::')

int main() {

int num; // Declare a variable to store the number entered by the user

// Ask the user to input a number

cout << "Enter a number: ";

cin >> num; // Read the number entered by the user and store it in 'num'

// Use the += operator to add 10 to the number

num += 10; // This is equivalent to: num = num + 10;

// Display the updated value of the number

cout << "The updated value after adding 10 is: " << num << endl;

return 0; // Return 0 to indicate successful execution of the program

}

SET 11

Q1: Write a program to check if a number entered by the user is positive, negative, or zero..(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

// Declare a variable to store the user input

float number;

// Prompt the user to enter a number

cout << "Enter a number: ";

cin >> number;

// Check if the number is positive, negative, or zero

if (number > 0) {

cout << "The number is positive." << endl;

} else if (number < 0) {

cout << "The number is negative." << endl;

} else {

cout << "The number is zero." << endl;

}

return 0;

}

Q2Write a program to check if a student passes an exam. A student passes if their marks are 40 or above.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int marks;

// Ask user for input marks

cout << "Enter the student's marks: ";

cin >> marks;

// Check if the student has passed

if (marks >= 40) {

cout << "The student has passed the exam!" << endl;

} else {

cout << "The student has failed the exam!" << endl;

}

return 0;

}

Q3: Write a program to input a character and check if it is a vowel (a, e, i, o, u) or a consonant using a switch statement.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

char ch;

// Ask user for input character

cout << "Enter a character: ";

cin >> ch;

// Convert the character to lowercase to handle both uppercase and lowercase inputs

ch = tolower(ch);

// Use switch case to check if the character is a vowel or consonant

switch (ch) {

case 'a':

case 'e':

case 'i':

case 'o':

case 'u':

cout << ch << " is a vowel." << endl;

break;

default:

cout << ch << " is a consonant." << endl;

}

return 0;

}

SET 12

Q1 : Write a program to input a number and print its multiplication table up to 10 using a for loop.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int number;

// Ask the user to input a number

cout << "Enter a number: ";

cin >> number;

// Loop to print multiplication table from 1 to 10

for (int i = 1; i <= 10; i++) {

cout << number << " x " << i << " = " << number \* i << endl;

}

return 0;

}

Q2 :Write a program to swap the values of two variables and display the result(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int a, b, temp;

// Ask user to input two numbers

cout << "Enter the value of a: ";

cin >> a;

cout << "Enter the value of b: ";

cin >> b;

// Swapping values using a temporary variable

temp = a;

a = b;

b = temp;

// Display the result after swapping

cout << "After swapping, the value of a is: " << a << endl;

cout << "After swapping, the value of b is: " << b << endl;

return 0;

}

Q3 Write a program to input a string and extract a substring from it (starting from a specified position for a specified length).(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

#include <string>

using namespace std;

int main() {

string str;

int start, length;

// Input the string

cout << "Enter a string: ";

getline(cin, str);

// Input the starting position and length of the substring

cout << "Enter the starting position: ";

cin >> start;

cout << "Enter the length of the substring: ";

cin >> length;

// Check if the starting position and length are valid

if (start >= 0 && start < str.length() && length > 0 && (start + length) <= str.length()) {

// Extract the substring using the substr() method

string substring = str.substr(start, length);

// Display the extracted substring

cout << "Extracted substring: " << substring << endl;

} else {

cout << "Invalid starting position or length." << endl;

}

return 0;

}

Set 13

Q1 Write a program to input two numbers and find the minimum using the min() function.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

#include <algorithm> // For the min() function

using namespace std;

int main() {

int num1, num2;

// Input two numbers

cout << "Enter the first number: ";

cin >> num1;

cout << "Enter the second number: ";

cin >> num2;

// Find the minimum using the min() function

int minimum = min(num1, num2);

// Output the minimum value

cout << "The minimum of " << num1 << " and " << num2 << " is: " << minimum << endl;

return 0;

}

Q2 Write a program to input some text from the user and save it to a text file.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

#include <fstream> // For file handling

#include <string>

using namespace std;

int main() {

string text;

ofstream outFile; // Create an output file stream object

// Ask user for the text input

cout << "Enter some text: ";

getline(cin, text); // Get the entire line of input, including spaces

// Open the file in write mode (it creates the file if it doesn't exist)

outFile.open("output.txt");

// Check if the file is opened successfully

if (outFile.is\_open()) {

// Write the input text to the file

outFile << text;

// Close the file after writing

outFile.close();

cout << "Text has been saved to 'output.txt'." << endl;

} else {

cout << "Error opening the file." << endl;

}

return 0;

}

Q3 Write a program to convert a given string to uppercase without using the built-in upper() function.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

#include <string>

using namespace std;

int main() {

string str;

// Ask the user for input string

cout << "Enter a string: ";

getline(cin, str);

// Convert each character to uppercase if it's a lowercase letter

for (int i = 0; i < str.length(); i++) {

if (str[i] >= 'a' && str[i] <= 'z') {

str[i] = str[i] - ('a' - 'A'); // Convert to uppercase

}

}

// Output the converted string

cout << "The string in uppercase is: " << str << endl;

return 0;

}

Set 14

Q1 Write a program to find whether a given year is a leap year.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int year;

// Input the year from the user

cout << "Enter a year: ";

cin >> year;

// Check if the year is a leap year

if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0)) {

cout << year << " is a leap year." << endl;

} else {

cout << year << " is not a leap year." << endl;

}

return 0;

}

Q2 :Write a program to input some text from the user and save it to a text file.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

#include <fstream> // For file handling

#include <string>

using namespace std;

int main() {

string text;

ofstream outFile; // Create an output file stream object

// Ask user for the text input

cout << "Enter some text: ";

getline(cin, text); // Get the entire line of input, including spaces

// Open the file in write mode (it creates the file if it doesn't exist)

outFile.open("output.txt");

// Check if the file is opened successfully

if (outFile.is\_open()) {

// Write the input text to the file

outFile << text;

// Close the file after writing

outFile.close();

cout << "Text has been saved to 'output.txt'." << endl;

} else {

cout << "Error opening the file." << endl;

}

return 0;

}

Q3 C:Write a program to read integers from a binary file and display them on the screen..(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

#include <fstream> // For file handling

using namespace std;

int main() {

ifstream inFile; // Create an input file stream object

int number;

// Open the binary file in input mode

inFile.open("numbers.bin", ios::in | ios::binary);

// Check if the file was opened successfully

if (!inFile) {

cout << "Error opening the file!" << endl;

return 1;

}

// Read and display the integers from the file

cout << "The integers read from the binary file are:" << endl;

while (inFile.read(reinterpret\_cast<char\*>(&number), sizeof(number))) {

cout << number << endl;

}

// Close the file

inFile.close();

return 0;

}

Set 15

Q1 : Write a program to convert a temperature from Celsius to Fahrenheit using the formula: F = \frac{9}{5}C + 32 .6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks

#include <iostream>

using namespace std;

int main() {

double celsius, fahrenheit;

// Input temperature in Celsius

cout << "Enter temperature in Celsius: ";

cin >> celsius;

// Convert Celsius to Fahrenheit

fahrenheit = (9.0 / 5.0) \* celsius + 32;

// Output the result

cout << celsius << " Celsius is equal to " << fahrenheit << " Fahrenheit." << endl;

return 0;

}

Q2 Write a program to calculate the average of five numbers entered by the user..(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks

#include <iostream>

using namespace std;

int main() {

double num1, num2, num3, num4, num5, average;

// Input five numbers

cout << "Enter five numbers: ";

cin >> num1 >> num2 >> num3 >> num4 >> num5;

// Calculate the average

average = (num1 + num2 + num3 + num4 + num5) / 5;

// Output the average

cout << "The average of the five numbers is: " << average << endl;

return 0;

}

Q3 : Write a program to input a number and use the -= operator to subtract 5 from the number, then display the updated value.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int number;

// Input a number from the user

cout << "Enter a number: ";

cin >> number;

// Subtract 5 from the number using the -= operator

number -= 5;

// Display the updated value

cout << "The updated value after subtracting 5 is: " << number << endl;

return 0;

}

Set 16

Q1 Write a program to calculate the sum of all numbers from 1 to n using recursion.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

// Recursive function to calculate sum from 1 to n

int sum(int n) {

// Base case: if n is 1, return 1

if (n == 1) {

return 1;

}

// Recursive case: sum of n is n + sum of n-1

return n + sum(n - 1);

}

int main() {

int n;

// Input the value of n

cout << "Enter a number n: ";

cin >> n;

// Call the recursive function to calculate the sum

int result = sum(n);

// Output the result

cout << "The sum of numbers from 1 to " << n << " is: " << result << endl;

return 0;

}

Q2 :Write a program to find the product of three numbers entered by the user.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

double num1, num2, num3, product;

// Input three numbers from the user

cout << "Enter three numbers: ";

cin >> num1 >> num2 >> num3;

// Calculate the product of the three numbers

product = num1 \* num2 \* num3;

// Output the product

cout << "The product of the three numbers is: " << product << endl;

return 0;

}

Q3 : Write a program to calculate the average of five numbers entered by the user.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

double num1, num2, num3, num4, num5, average;

// Input five numbers from the user

cout << "Enter five numbers: ";

cin >> num1 >> num2 >> num3 >> num4 >> num5;

// Calculate the average

average = (num1 + num2 + num3 + num4 + num5) / 5;

// Output the average

cout << "The average of the five numbers is: " << average << endl;

return 0;

}

Set 17

Q1 Write a program to check if a person is eligible to vote. The person must be at least 18 years old and a citizen of the country.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

#include <string>

using namespace std;

int main() {

int age;

string citizenship;

// Input the age and citizenship status

cout << "Enter your age: ";

cin >> age;

cout << "Are you a citizen of the country? (yes/no): ";

cin >> citizenship;

// Check if the person is eligible to vote

if (age >= 18 && (citizenship == "yes" || citizenship == "Yes")) {

cout << "You are eligible to vote." << endl;

} else {

cout << "You are not eligible to vote." << endl;

}

return 0;

}

Q2 : Write a program to check if a given number lies within a specific range (e.g., between 10 and 50 inclusive)..(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int num;

// Input a number from the user

cout << "Enter a number: ";

cin >> num;

// Check if the number lies within the range 10 to 50 inclusive

if (num >= 10 && num <= 50) {

cout << "The number is within the range of 10 to 50." << endl;

} else {

cout << "The number is outside the range of 10 to 50." << endl;

}

return 0;

}

Q3 : Write a program to input two integers and display their quotient and remainder.(6Marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int num1, num2;

// Input two integers from the user

cout << "Enter two integers: ";

cin >> num1 >> num2;

// Check if the second number is not zero to avoid division by zero

if (num2 != 0) {

// Calculate the quotient and remainder

int quotient = num1 / num2;

int remainder = num1 % num2;

// Display the quotient and remainder

cout << "Quotient: " << quotient << endl;

cout << "Remainder: " << remainder << endl;

} else {

cout << "Error: Division by zero is not allowed!" << endl;

}

return 0;

}

Set 18

Q1 : Write a program to calculate the factorial of a number using iteration.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int num;

long long factorial = 1; // Use long long to handle large numbers

// Input a number from the user

cout << "Enter a number: ";

cin >> num;

// Check if the number is negative

if (num < 0) {

cout << "Factorial is not defined for negative numbers." << endl;

} else {

// Calculate factorial using iteration

for (int i = 1; i <= num; i++) {

factorial \*= i;

}

// Output the factorial

cout << "The factorial of " << num << " is: " << factorial << endl;

}

return 0;

}

Q2 : Write a program to calculate the average of five numbers entered by the user.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

double num1, num2, num3, num4, num5, average;

// Input five numbers from the user

cout << "Enter five numbers: ";

cin >> num1 >> num2 >> num3 >> num4 >> num5;

// Calculate the average

average = (num1 + num2 + num3 + num4 + num5) / 5;

// Output the average

cout << "The average of the five numbers is: " << average << endl;

return 0;

}

Q3 Write a program to check if a number entered by the user is positive, negative, or zero.6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int num;

// Input a number from the user

cout << "Enter a number: ";

cin >> num;

// Check if the number is positive, negative, or zero

if (num > 0) {

cout << "The number is positive." << endl;

} else if (num < 0) {

cout << "The number is negative." << endl;

} else {

cout << "The number is zero." << endl;

}

return 0;

}

Set 19

Q1 : Write a program to check if a year entered by the user is a leap year.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int year;

// Input the year from the user

cout << "Enter a year: ";

cin >> year;

// Check if the year is a leap year

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

cout << year << " is a leap year." << endl;

} else {

cout << year << " is not a leap year." << endl;

}

return 0;

}

Q2 Write a program to input a number (1 to 7) and display the corresponding day of the week using a switch statement..(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int day;

// Input a number between 1 and 7 from the user

cout << "Enter a number (1 to 7): ";

cin >> day;

// Display the corresponding day of the week using a switch statement

switch (day) {

case 1:

cout << "Monday" << endl;

break;

case 2:

cout << "Tuesday" << endl;

break;

case 3:

cout << "Wednesday" << endl;

break;

case 4:

cout << "Thursday" << endl;

break;

case 5:

cout << "Friday" << endl;

break;

case 6:

cout << "Saturday" << endl;

break;

case 7:

cout << "Sunday" << endl;

break;

default:

cout << "Invalid input! Please enter a number between 1 and 7." << endl;

}

return 0;

}

Q3 Write a program to input a number (1 to 12) and display the corresponding month name using a switch statement (6Marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks

#include <iostream>

using namespace std;

int main() {

int month;

// Input a number between 1 and 12 from the user

cout << "Enter a number (1 to 12): ";

cin >> month;

// Display the corresponding month name using a switch statement

switch (month) {

case 1:

cout << "January" << endl;

break;

case 2:

cout << "February" << endl;

break;

case 3:

cout << "March" << endl;

break;

case 4:

cout << "April" << endl;

break;

case 5:

cout << "May" << endl;

break;

case 6:

cout << "June" << endl;

break;

case 7:

cout << "July" << endl;

break;

case 8:

cout << "August" << endl;

break;

case 9:

cout << "September" << endl;

break;

case 10:

cout << "October" << endl;

break;

case 11:

cout << "November" << endl;

break;

case 12:

cout << "December" << endl;

break;

default:

cout << "Invalid input! Please enter a number between 1 and 12." << endl;

}

return 0;

}

Set 20

Q1 A: Write a program to check if a number entered by the user is positive, negative, or zero. or not.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks)

#include <iostream>

using namespace std;

int main() {

int num;

// Input a number from the user

cout << "Enter a number: ";

cin >> num;

// Check if the number is positive, negative, or zero

if (num > 0) {

cout << "The number is positive." << endl;

} else if (num < 0) {

cout << "The number is negative." << endl;

} else {

cout << "The number is zero." << endl;

}

return 0;

}

Q2 Write a program to find the largest number in 3 numbers(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks

#include <iostream>

using namespace std;

int main() {

int num1, num2, num3;

// Input three numbers from the user

cout << "Enter three numbers: ";

cin >> num1 >> num2 >> num3;

// Determine the largest number

if (num1 >= num2 && num1 >= num3) {

cout << "The largest number is: " << num1 << endl;

} else if (num2 >= num1 && num2 >= num3) {

cout << "The largest number is: " << num2 << endl;

} else {

cout << "The largest number is: " << num3 << endl;

}

return 0;

}

Q3 Write a program to print the multiplication table of a number.(6 marks) Write the breakdown for the above program (2 marks) Give 2 possible input/ output cases (2 marks

#include <iostream>

using namespace std;

int main() {

int num;

// Input the number from the user

cout << "Enter a number to print its multiplication table: ";

cin >> num;

// Print the multiplication table for the entered number

cout << "Multiplication table of " << num << " is:" << endl;

for (int i = 1; i <= 10; i++) {

cout << num << " x " << i << " = " << num \* i << endl;

}

return 0;

}

Set - 21

1. Write a program to input a number (1 to 7) and display the corresponding day of the

week using a switch statement

#include <iostream> // Include the input-output stream library to use 'cin' and 'cout'

using namespace std; // Use the standard namespace to avoid using 'std::' before 'cin' and 'cout'

int main() {

int day; // Declare an integer variable 'day' to store the user's input (1 to 7)

cout << "Enter a number (1 to 7): "; // Prompt the user to enter a number

cin >> day; // Take input from the user and store it in the variable 'day'

switch(day) {

case 1: // If the input is 1

cout << "Sunday"; // Output "Sunday"

break; // Exit the switch statement to avoid fall-through to other cases

case 2: // If the input is 2

cout << "Monday"; // Output "Monday"

break; // Exit the switch statement

case 3: // If the input is 3

cout << "Tuesday"; // Output "Tuesday"

break; // Exit the switch statement

case 4: // If the input is 4

cout << "Wednesday"; // Output "Wednesday"

break; // Exit the switch statement

case 5: // If the input is 5

cout << "Thursday"; // Output "Thursday"

break; // Exit the switch statement

case 6: // If the input is 6

cout << "Friday"; // Output "Friday"

break; // Exit the switch statement

case 7: // If the input is 7

cout << "Saturday"; // Output "Saturday"

break; // Exit the switch statement

default: // If the input is not between 1 and 7

cout << "Invalid input! Please enter a number between 1 and 7."; // Display an error message

}

return 0; // Return 0 to indicate that the program has executed successfully

}

1. Write a program to check if a number is divisible by 5 and 3

#include <iostream> // Include the header file for input-output operations

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

int main() {

int num; // Declare an integer variable 'num' to store the user input number

// Prompt the user to enter a number

cout << "Enter a number: ";

cin >> num; // Input the number from the user

// Check if the number is divisible by both 5 and 3

if (num % 5 == 0 && num % 3 == 0) {

// If the number is divisible by both 5 and 3

cout << num << " is divisible by both 5 and 3." << endl;

} else {

// If the number is not divisible by both 5 and 3

cout << num << " is not divisible by both 5 and 3." << endl;

}

return 0; // Return 0 to indicate successful execution of the program

}

1. Write a program to check whether a number is even or odd

#include <iostream> // Include the input-output stream library for cin and cout

using namespace std; // Use the standard namespace to avoid writing 'std::' before cin and cout

int main() {

int num; // Declare an integer variable 'num' to store the user's input

// Ask the user to enter a number

cout << "Enter a number: ";

cin >> num; // Input the number from the user and store it in 'num'

// Check if the number is even or odd using the modulus operator

if (num % 2 == 0) { // If the remainder when 'num' is divided by 2 is 0, the number is even

cout << num << " is even." << endl; // Output that the number is even

} else { // If the remainder is not 0, the number is odd

cout << num << " is odd." << endl; // Output that the number is odd

}

return 0; // Return 0 to indicate successful execution of the program

}

Set -22

1. Write a program to declare variables for your name, age, and country, and print them.

#include <iostream> // Include the header file for input-output operations

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

int main() {

// Declare variables to store name, age, and country

string name = "John Doe"; // String variable to store name

int age = 25; // Integer variable to store age

string country = "USA"; // String variable to store country

// Output the values of name, age, and country

cout << "Name: " << name << endl; // Print the name

cout << "Age: " << age << endl; // Print the age

cout << "Country: " << country << endl; // Print the country

return 0; // Return 0 to indicate successful execution of the program

}

1. :Write a program to check if a person is eligible to vote. The person must be at least 18 years old and a citizen of the country

#include <iostream> // Include the header file for input-output operations

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

int main() {

int age; // Declare an integer variable 'age' to store the person's age

string citizen; // Declare a string variable 'citizen' to store the citizenship status

// Ask the user for their age

cout << "Enter your age: ";

cin >> age; // Input the age from the user and store it in 'age'

// Ask the user if they are a citizen

cout << "Are you a citizen of the country? (yes/no): ";

cin >> citizen; // Input the citizenship status from the user and store it in 'citizen'

// Check if the person is eligible to vote

if (age >= 18 && (citizen == "yes" || citizen == "Yes")) {

// If the person is 18 or older and is a citizen

cout << "You are eligible to vote." << endl;

} else {

// If the person is not 18 or older, or not a citizen

cout << "You are not eligible to vote." << endl;

}

return 0; // Return 0 to indicate successful execution of the program

}

1. Write a program to input a number and find its square root using the sqrt() function

#include <iostream> // Include the input-output stream library for cin and cout

#include <cmath> // Include the cmath library to use the sqrt() function

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

int main() {

double num; // Declare a variable 'num' to store the input number

double result; // Declare a variable 'result' to store the square root

// Ask the user to enter a number

cout << "Enter a number: ";

cin >> num; // Input the number from the user and store it in 'num'

// Check if the number is non-negative

if (num < 0) {

// If the number is negative, square root is not defined for real numbers

cout << "Error: Cannot compute the square root of a negative number." << endl;

} else {

// Calculate the square root of the number using sqrt() function

result = sqrt(num); // sqrt() returns the square root of 'num'

cout << "The square root of " << num << " is: " << result << endl; // Output the result

}

return 0; // Return 0 to indicate successful execution of the program

}

Set -23

1. Write a program to define a class Car with a constructor to initialize brand and model. Create an object and display these values

#include <iostream> // Include the header file for input-output operations

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

// Define the Car class

class Car {

// Private member variables to store the brand and model of the car

string brand;

string model;

public:

// Constructor to initialize brand and model

Car(string b, string m) {

brand = b; // Initialize the brand of the car

model = m; // Initialize the model of the car

}

// Function to display the brand and model of the car

void display() {

cout << "Car Brand: " << brand << endl; // Print the car brand

cout << "Car Model: " << model << endl; // Print the car model

}

};

// Main function

int main() {

// Create an object of the Car class and initialize it with brand and model

Car myCar("Toyota", "Corolla");

// Call the display function to show the brand and model of the car

myCar.display();

return 0; // Return 0 to indicate successful execution of the program

}

1. : Write a program to input some integers from the user and save them to a binary file

#include <iostream> // Include the header file for input-output operations

#include <fstream> // Include the header file for file handling (file streams)

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

int main() {

int n; // Declare a variable to store the number of integers to be input

cout << "Enter the number of integers you want to input: ";

cin >> n; // Input the number of integers from the user

// Open a binary file to write the integers

ofstream outFile("numbers.dat", ios::binary); // Open the file "numbers.dat" in binary write mode

// Check if the file was opened successfully

if (!outFile) {

cout << "Error opening file!" << endl; // Display an error message if the file couldn't be opened

return 1; // Return 1 to indicate an error occurred

}

// Loop to input 'n' integers from the user and write them to the binary file

cout << "Enter " << n << " integers: " << endl;

for (int i = 0; i < n; ++i) {

int num; // Declare a variable to store each integer

cout << "Enter integer " << (i + 1) << ": ";

cin >> num; // Input an integer from the user

outFile.write(reinterpret\_cast<char\*>(&num), sizeof(num)); // Write the integer to the binary file

}

// Close the file after writing

outFile.close();

cout << "Integers have been written to the binary file successfully." << endl; // Display a success message

return 0; // Return 0 to indicate successful execution of the program

}

1. : Write a program to input a number and find its square root using the sqrt() function.

#include <iostream> // Include the header file for input-output operations

#include <cmath> // Include the cmath library to use the sqrt() function

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

int main() {

double num; // Declare a variable to store the input number

double result; // Declare a variable to store the square root result

// Ask the user to enter a number

cout << "Enter a number: ";

cin >> num; // Read the input number from the user

// Check if the number is negative

if (num < 0) {

// If the number is negative, square root is not defined for real numbers

cout << "Error: Cannot compute the square root of a negative number." << endl;

} else {

// Calculate the square root of the number using sqrt() function

result = sqrt(num); // sqrt() returns the square root of num

cout << "The square root of " << num << " is: " << result << endl; // Display the result

}

return 0; // Return 0 to indicate successful execution of the program

}

Set – 24

1. Write a program to find whether a number is divisible by 3 and 5

#include <iostream> // Include the header file for input-output operations

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

int main() {

int num; // Declare a variable to store the number

// Ask the user to input a number

cout << "Enter a number: ";

cin >> num; // Read the number from the user

// Check if the number is divisible by both 3 and 5

if (num % 3 == 0 && num % 5 == 0) {

cout << num << " is divisible by both 3 and 5." << endl; // Display message if divisible by both

} else {

cout << num << " is not divisible by both 3 and 5." << endl; // Display message if not divisible by both

}

return 0; // Return 0 to indicate successful execution of the program

}

1. Write a program to calculate the circumference and area of a circle given its radius.

Use \pi = 3.14

#include <iostream> // Include the header file for input-output operations

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

int main() {

const double PI = 3.14; // Define a constant for Pi (π) with value 3.14

double radius, circumference, area; // Declare variables for radius, circumference, and area

// Ask the user to enter the radius of the circle

cout << "Enter the radius of the circle: ";

cin >> radius; // Read the radius from the user

// Calculate the circumference of the circle: C = 2 \* π \* radius

circumference = 2 \* PI \* radius; // Formula for circumference

// Calculate the area of the circle: A = π \* radius^2

area = PI \* radius \* radius; // Formula for area

// Display the results

cout << "Circumference of the circle: " << circumference << endl; // Print the circumference

cout << "Area of the circle: " << area << endl; // Print the area

return 0; // Return 0 to indicate successful execution of the program

}

1. Write a program to check if a password is valid. A password is valid if it has at least 8 characters and does not contain any spaces.

#include <iostream> // Include the header file for input-output operations

#include <string> // Include the string library to use the string type and functions

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

int main() {

string password; // Declare a string variable to store the password

// Ask the user to input a password

cout << "Enter your password: ";

getline(cin, password); // Read the entire password (including spaces) from the user

// Check if the password is valid (length >= 8 and no spaces)

if (password.length() >= 8 && password.find(' ') == string::npos) {

// If the password is valid, print a success message

cout << "Password is valid." << endl;

} else {

// If the password is not valid, print an error message

cout << "Password is invalid." << endl;

}

return 0; // Return 0 to indicate successful execution of the program

}

Set – 25

1. :Write a program to input a number and find its square root using the sqrt() function

#include <iostream> // Include the header file for input-output operations

#include <cmath> // Include the cmath library to use the sqrt() function

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

int main() {

double number, result; // Declare variables to store the input number and the result of square root

// Ask the user to input a number

cout << "Enter a number: ";

cin >> number; // Read the number from the user

// Calculate the square root using the sqrt() function

result = sqrt(number); // sqrt() returns the square root of the given number

// Display the result

cout << "The square root of " << number << " is " << result << endl; // Print the result

return 0; // Return 0 to indicate successful execution of the program

}

1. : Write a program to demonstrate simple inheritance by creating a base class Animal with an attribute name and a derived class Dog that adds a method bark()

#include <iostream> // Include the header file for input-output operations

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

// Base class Animal

class Animal {

public:

string name; // Attribute to store the name of the animal

// Constructor to initialize the name of the animal

Animal(string animalName) {

name = animalName;

}

// Method to display the name of the animal

void displayName() {

cout << "The animal's name is: " << name << endl;

}

};

// Derived class Dog that inherits from Animal

class Dog : public Animal {

public:

// Constructor to initialize the name of the dog by calling the base class constructor

Dog(string dogName) : Animal(dogName) {}

// Method specific to the Dog class

void bark() {

cout << name << " says: Woof Woof!" << endl;

}

};

int main() {

// Create an object of the Dog class

Dog myDog("Buddy");

// Call the displayName() method from the base class

myDog.displayName();

// Call the bark() method from the derived class

myDog.bark();

return 0; // Return 0 to indicate successful execution of the program

}

1. :Write a program to input some text from the user and save it to a text file..*(6 marks)*

*Write the breakdown for the above program*

#include <iostream> // Include the header file for input-output operations

#include <fstream> // Include the header file for file handling (reading/writing files)

#include <string> // Include the header file to use string type

using namespace std; // Use the standard namespace to avoid using 'std::' before cin and cout

int main() {

string userInput; // Declare a string variable to store the user's input

ofstream outputFile("output.txt"); // Create an output file stream object and open "output.txt" for writing

// Check if the file was opened successfully

if (!outputFile) {

cout << "Error opening file!" << endl; // If the file could not be opened, display an error message

return 1; // Return 1 to indicate an error occurred

}

// Ask the user to input some text

cout << "Enter some text: ";

getline(cin, userInput); // Read the entire line of text entered by the user, including spaces

// Write the user input to the text file

outputFile << userInput << endl; // Write the input text to the file followed by a newline

cout << "Text has been saved to the file." << endl; // Inform the user that the text was saved

outputFile.close(); // Close the file to ensure that all data is written and resources are released

return 0; // Return 0 to indicate the program executed successfully

}

Set – 26

1. Write a program to swap the values of two variables and display the result

#include <iostream> // Include the input-output stream library

using namespace std; // Allow using standard library objects without prefixing 'std::'

int main() {

// Declare two integer variables and initialize them

int a, b;

// Ask user to input values for a and b

cout << "Enter the value of a: ";

cin >> a; // Read value of a from user

cout << "Enter the value of b: ";

cin >> b; // Read value of b from user

// Display values before swapping

cout << "Before swapping:" << endl;

cout << "a = " << a << ", b = " << b << endl;

// Swap the values of a and b using a temporary variable

int temp = a; // Store the value of a in temp

a = b; // Assign the value of b to a

b = temp; // Assign the value of temp (old value of a) to b

// Display values after swapping

cout << "After swapping:" << endl;

cout << "a = " << a << ", b = " << b << endl;

return 0; // End of the program

}

1. Write a program to demonstrate simple inheritance by creating a base class Animal with an attribute name and a derived class Dog that adds a method bark()

#include <iostream> // Include the input-output stream library

using namespace std; // Use the standard library (no need to prefix 'std::')

// Base class Animal

class Animal {

public:

// Declare an attribute 'name' to store the name of the animal

string name;

// Constructor to initialize the name of the animal

Animal(string animalName) {

name = animalName; // Initialize 'name' with the provided animal name

}

// Method to display the name of the animal

void display() {

cout << "Animal name: " << name << endl; // Display the animal's name

}

};

// Derived class Dog inherits from the Animal class

class Dog : public Animal {

public:

// Constructor to initialize the name of the dog

Dog(string dogName) : Animal(dogName) {

// The constructor of the base class Animal is called here

}

// Method to make the dog bark

void bark() {

cout << name << " says: Woof Woof!" << endl; // Display a barking message

}

};

int main() {

// Create an object of the derived class Dog

Dog myDog("Buddy"); // Initialize the dog's name as "Buddy"

// Call methods from both the base class Animal and the derived class Dog

myDog.display(); // Display the name of the animal (in this case, the dog)

myDog.bark(); // Make the dog bark

return 0; // End of the program

}

1. Write a program to input some integers from the user and save them to a binary file

#include <iostream> // Include the input-output stream library

#include <fstream> // Include the file stream library to handle file operations

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

int n; // Declare an integer variable 'n' to store the number of integers to be entered

// Ask the user for the number of integers they want to input

cout << "Enter the number of integers you want to save to the file: ";

cin >> n; // Read the user's input into 'n'

// Open a binary file for writing. The file will be created if it doesn't exist.

ofstream outFile("integers.dat", ios::binary);

// Check if the file was opened successfully

if (!outFile) {

cout << "Error opening file for writing." << endl; // If the file can't be opened, display an error message

return 1; // Exit the program with an error code

}

// Input the integers and write them to the file

cout << "Enter " << n << " integers: " << endl;

// Loop through 'n' times to input integers

for (int i = 0; i < n; i++) {

int num; // Declare a variable 'num' to store each integer

cout << "Enter integer " << (i + 1) << ": ";

cin >> num; // Read the integer from the user

// Write the integer to the binary file

outFile.write(reinterpret\_cast<char\*>(&num), sizeof(num));

}

// Close the file after writing

outFile.close();

cout << "Integers have been successfully saved to 'integers.dat'." << endl; // Inform the user that the integers are saved

return 0; // End the program successfully

}

Set – 27

1. Write a program to define a class Car with a constructor to initialize brand and model. Create an object and display these values

#include <iostream> // Include the input-output stream library to handle user input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

// Define the class Car

class Car {

public:

string brand; // Declare a string variable to store the brand of the car

string model; // Declare a string variable to store the model of the car

// Constructor to initialize the brand and model of the car

Car(string b, string m) {

brand = b; // Initialize 'brand' with the value of 'b'

model = m; // Initialize 'model' with the value of 'm'

}

// Method to display the brand and model of the car

void display() {

cout << "Car Brand: " << brand << endl; // Display the brand of the car

cout << "Car Model: " << model << endl; // Display the model of the car

}

};

int main() {

// Create an object of the Car class with the brand "Toyota" and model "Corolla"

Car myCar("Toyota", "Corolla");

// Call the display method to show the car's brand and model

myCar.display(); // This will print the brand and model of the car

return 0; // End of the program

}

1. Write a program to convert a temperature from Celsius to Fahrenheit using the formula: F = \frac{9}{5}C + 32

#include <iostream> // Include the input-output stream library to handle user input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

float celsius, fahrenheit; // Declare variables to store temperature in Celsius and Fahrenheit

// Ask the user to input a temperature in Celsius

cout << "Enter temperature in Celsius: ";

cin >> celsius; // Read the temperature in Celsius from the user

// Apply the formula to convert Celsius to Fahrenheit

fahrenheit = (9.0 / 5.0) \* celsius + 32; // Convert the temperature using the formula

// Display the result: the temperature in Fahrenheit

cout << "Temperature in Fahrenheit: " << fahrenheit << "°F" << endl;

return 0; // End of the program

}

1. Write a program to check if a password is valid. A password is valid if it has at least 8

characters and does not contain any spaces.

#include <iostream> // Include the input-output stream library for user input and output

#include <string> // Include the string library to handle string operations

using namespace std; // Use the standard library (no need to prefix 'std::')

bool isValidPassword(const string& password) {

// Check if the password length is at least 8 characters

if (password.length() < 8) {

return false; // Return false if the password length is less than 8

}

// Check if the password contains any spaces

if (password.find(' ') != string::npos) {

return false; // Return false if a space is found in the password

}

return true; // Return true if the password meets both conditions

}

int main() {

string password; // Declare a string variable to store the user input for the password

// Ask the user to enter a password

cout << "Enter a password: ";

cin >> password; // Read the password entered by the user

// Check if the entered password is valid using the isValidPassword function

if (isValidPassword(password)) {

cout << "Password is valid!" << endl; // If valid, print this message

} else {

cout << "Password is invalid. It must be at least 8 characters long and not contain spaces." << endl; // If invalid, print this message

}

return 0; // End of the program

}

Set – 28

1. Write a program to demonstrate simple inheritance by creating a base class Animal

with an attribute name and a derived class Dog that adds a method bark()

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

// Define the base class Animal

class Animal {

public:

string name; // Declare a public attribute 'name' to store the name of the animal

// Constructor to initialize the name of the animal

Animal(string n) {

name = n; // Initialize 'name' with the value of 'n'

}

// A method to display the animal's name

void displayName() {

cout << "Animal's Name: " << name << endl; // Display the name of the animal

}

};

// Define the derived class Dog that inherits from the Animal class

class Dog : public Animal {

public:

// Constructor to initialize the name of the dog using the base class constructor

Dog(string n) : Animal(n) { }

// A method specific to the Dog class that makes the dog bark

void bark() {

cout << name << " says: Woof!" << endl; // Display the barking sound

}

};

int main() {

// Create an object of the Dog class with the name "Buddy"

Dog myDog("Buddy");

// Call the displayName method inherited from the Animal class

myDog.displayName(); // This will display "Buddy"

// Call the bark method specific to the Dog class

myDog.bark(); // This will make Buddy bark: "Woof!"

return 0; // End of the program

}

1. Write a program to input some text from the user and save it to a text file.

#include <iostream> // Include the input-output stream library for reading input and writing output

#include <fstream> // Include the file stream library to work with files

#include <string> // Include the string library to handle string operations

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

string userText; // Declare a string variable to store the input text

// Ask the user to enter some text

cout << "Enter some text: ";

getline(cin, userText); // Read a full line of text (including spaces) from the user input

// Create and open a text file for writing (output file stream)

ofstream outputFile("output.txt");

// Check if the file was successfully opened

if (outputFile.is\_open()) {

// Write the user input text to the file

outputFile << userText << endl; // Write the entered text followed by a newline

// Inform the user that the text has been saved to the file

cout << "Text has been saved to output.txt" << endl;

// Close the file after writing

outputFile.close();

} else {

// If the file couldn't be opened, display an error message

cout << "Error opening the file!" << endl;

}

return 0; // End of the program

}

1. Write a program to input a number and find its square root using the sqrt() function.

#include <iostream> // Include the input-output stream library to handle input and output

#include <cmath> // Include the cmath library to access mathematical functions like sqrt()

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

double number; // Declare a variable to store the number entered by the user

// Ask the user to input a number

cout << "Enter a number: ";

cin >> number; // Read the user's input into the 'number' variable

// Check if the number is non-negative, because square root of negative numbers is not real

if (number < 0) {

cout << "Error: Cannot calculate square root of a negative number!" << endl;

} else {

// Calculate the square root using the sqrt() function

double squareRoot = sqrt(number);

// Display the result

cout << "The square root of " << number << " is " << squareRoot << endl;

}

return 0; // End of the program

}

Set – 29

1. Write a program to calculate the circumference and area of a circle given its radius. Use \pi = 3.14

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

double radius; // Declare a variable to store the radius of the circle

double pi = 3.14; // Define the constant value of pi (3.14)

double circumference, area; // Declare variables to store the circumference and area of the circle

// Ask the user to input the radius of the circle

cout << "Enter the radius of the circle: ";

cin >> radius; // Read the radius value entered by the user

// Calculate the circumference using the formula: Circumference = 2 \* pi \* radius

circumference = 2 \* pi \* radius;

// Calculate the area using the formula: Area = pi \* radius^2

area = pi \* radius \* radius;

// Output the results: Circumference and Area

cout << "The circumference of the circle is: " << circumference << endl;

cout << "The area of the circle is: " << area << endl;

return 0; // End of the program

}

1. : Write a program to calculate the simple interest using the formula:

**SI = (P × R × T) / 100**

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

double principal, rate, time, simpleInterest; // Declare variables to store principal, rate, time, and simple interest

// Ask the user to input the principal amount

cout << "Enter the principal amount: ";

cin >> principal; // Read the principal value entered by the user

// Ask the user to input the rate of interest

cout << "Enter the rate of interest (in %): ";

cin >> rate; // Read the rate of interest value entered by the user

// Ask the user to input the time period (in years)

cout << "Enter the time period (in years): ";

cin >> time; // Read the time value entered by the user

// Calculate the simple interest using the formula: SI = (P × R × T) / 100

simpleInterest = (principal \* rate \* time) / 100; // Perform the calculation

// Display the calculated simple interest

cout << "The simple interest is: " << simpleInterest << endl;

return 0; // End of the program

}

1. :Write a program to swap the values of two variables and display the result

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

int a, b, temp; // Declare three integer variables: 'a', 'b', and 'temp' (used for swapping)

// Ask the user to input the first number

cout << "Enter the first number (a): ";

cin >> a; // Read the value of 'a' from user input

// Ask the user to input the second number

cout << "Enter the second number (b): ";

cin >> b; // Read the value of 'b' from user input

// Display the values before swapping

cout << "Before swapping: " << endl;

cout << "a = " << a << ", b = " << b << endl; // Print the values of 'a' and 'b'

// Swap the values of 'a' and 'b' using a temporary variable 'temp'

temp = a; // Store the value of 'a' in 'temp'

a = b; // Assign the value of 'b' to 'a'

b = temp; // Assign the value of 'temp' (original 'a') to 'b'

// Display the values after swapping

cout << "After swapping: " << endl;

cout << "a = " << a << ", b = " << b << endl; // Print the swapped values of 'a' and 'b'

return 0; // End of the program

}

Set – 30

1. Write a program to input a number and find its square root using the sqrt() function

#include <iostream> // Include the input-output stream library to handle input and output

#include <cmath> // Include the cmath library to access mathematical functions like sqrt()

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

double number, squareRoot; // Declare variables to store the input number and its square root

// Ask the user to input a number

cout << "Enter a number: ";

cin >> number; // Read the input number from the user

// Check if the number is non-negative, because sqrt() does not support negative numbers

if (number < 0) {

cout << "Error: Cannot calculate the square root of a negative number!" << endl;

} else {

// Calculate the square root using the sqrt() function from cmath library

squareRoot = sqrt(number); // Calculate the square root of the number

// Display the result

cout << "The square root of " << number << " is " << squareRoot << endl;

}

return 0; // End of the program

}

1. Write a program to input a number and use the += operator to add 10 to the number, then display the updated value

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

int number; // Declare a variable to store the number entered by the user

// Ask the user to input a number

cout << "Enter a number: ";

cin >> number; // Read the number entered by the user and store it in the 'number' variable

// Use the += operator to add 10 to the number

number += 10; // This is equivalent to number = number + 10;

// Display the updated value of the number

cout << "The updated number after adding 10 is: " << number << endl;

return 0; // End of the program

}

1. Write a program to check if a given number lies within a specific range (e.g., between 10 and 50 inclusive)

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

int number; // Declare a variable to store the number entered by the user

// Ask the user to input a number

cout << "Enter a number: ";

cin >> number; // Read the input number entered by the user

// Check if the number lies between 10 and 50 inclusive

if (number >= 10 && number <= 50) {

// If the condition is true, the number is within the range

cout << "The number " << number << " is within the range of 10 and 50." << endl;

} else {

// If the condition is false, the number is outside the range

cout << "The number " << number << " is outside the range of 10 and 50." << endl;

}

return 0; // End of the program

}

Set- 6

1. : Write a program to declare variables for your name, age, and country, and print them.

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard namespace, so we don't need to prefix with 'std::'

x

int main() {

string name = "John Doe"; // Declare a string variable 'name' and initialize it with "John Doe"

int age = 25; // Declare an integer variable 'age' and initialize it with the value 25

string country = "USA"; // Declare a string variable 'country' and initialize it with "USA"

// Print the name, age, and country

cout << "Name: " << name << endl; // Print the value of the 'name' variable

cout << "Age: " << age << endl; // Print the value of the 'age' variable

cout << "Country: " << country << endl; // Print the value of the 'country' variable

return 0; // End of the program, return 0 to indicate successful execution

}

1. :Write a program to convert a temperature from Celsius to Fahrenheit using the formula:

F = \frac{9}{5}C + 32

#include <iostream> // Include the input-output stream library to handle input and output

using namespace std; // Use the standard library (no need to prefix 'std::')

int main() {

double celsius, fahrenheit; // Declare variables to store Celsius and Fahrenheit values

// Ask the user to input a temperature in Celsius

cout << "Enter the temperature in Celsius: ";

cin >> celsius; // Read the Celsius temperature from the user

// Convert Celsius to Fahrenheit using the formula: F = (9/5) \* C + 32

fahrenheit = (9.0 / 5.0) \* celsius + 32; // Perform the conversion and store the result in 'fahrenheit'

// Display the converted temperature in Fahrenheit

cout << "The temperature in Fahrenheit is: " << fahrenheit << endl;

return 0; // End of the program

}