**EXPERIMENTNO.01**

**TITLE:-Complex NumberCalculator**

**Problem Statement:** - Implement a class complex which represents the complex number data type.Implement thefollowing

1. Constructor(includingadefaultconstructorwhichcreatesthecomplexnumber0+0i)
2. OverloadOperator+toaddtwo complexnumber
3. OverloadOperator\*tomultiplytwocomplexnumber.
4. OverloadOperator<< and>>toprintandreadcomplex number.

**SoftwareUsed**:-DevC++

**Competency/Skills:**

Thestudents willbe abletoimprovethefollowing skills

1. To understand and apply the concepts like inheritance, polymorphism, exception handlingandgeneric structurefor implementingreusable programmingcodes.
2. To analyze the concepts of file and apply it while storing and retrieving the data fromsecondarystorages
3. Tolearnbasics,featuresand futureofC++programming

**LearningObjectives:**

Thestudent should beable to:

1. Tolearnproblemsolvingwithcomputers
2. Tounderstandand developalgorithmsand implementprogramsusingC++language.
3. Tolearnfeatures ofObjectOrientedProgrammingusingC++programming.

**ExpectedOutcomes:**

Thestudentsshouldacquirethefollowingskill bythecompletion ofthisexperience:

1. Inculcateandapplyvariousskillsincomputergraphics
2. Exhibit the programming skills for the problems those require the writing of welldocumentedprogramsincludinguseofthelogicalconstructsoflanguage,C++.

**Practicalsignificance:**

In object oriented programming there are various methods are used to write programs, usingthis experiment the concepts of complex number operations and different

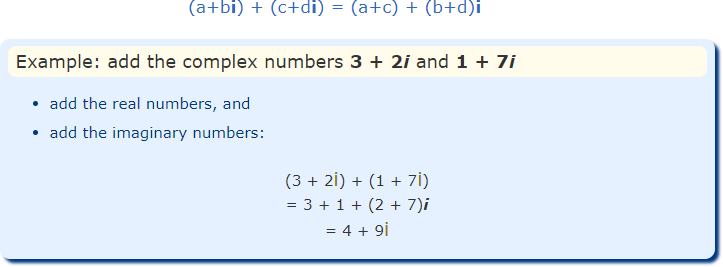
operators can beimplemented.

**Theory**:-

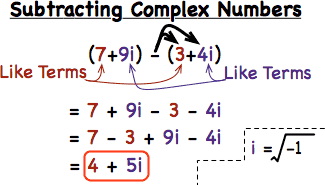
A **complexnumber** isa **number** thatcanbeexpressedintheforma+bi,whereaandbarereal**numbers**,and irepresentsthe imaginary unit, satisfyingthe equation i2 =−1.

Forthe complexnumber*a*+*bi*, *a*is called the**realpart**,and *b*iscalled the**imaginarypart**.

1. Addition

When adding complex numbers, add the real parts together and add the imaginary parttogether

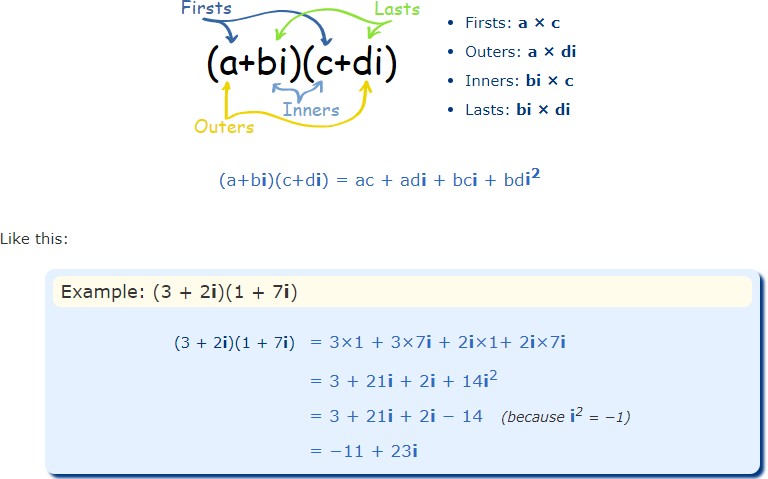
1. Subtraction

Whensubtractingcomplexnumber,subtracttherealparttogetherandsubtractimaginarypart together

1. Multiplication

Each part of the first complex numbergets multiplied byeachpartofthesecond

complex number



**FeaturesofTool/Language:**

1. **Simple**

C++ provides a structured approach wherein you can break the problem into parts anddesign the solution modularly. It provides you a rich set of library functions that youcan use while implementing the solution. If you have worked with [C language](https://www.edureka.co/blog/c-programming-tutorial/), thenmoving to C++ would be a very smooth transitioning. The syntax is almost similarwithminutechanges.

1. **PlatformDependent**

Platform dependent language means the language in which programs can be executedonly on that operating system where it is developed & compiled. It cannot run orexecuteit on any other operating system.

C++isaplatform-dependentlanguage.Havingsaidthat, [C++programs](https://www.edureka.co/blog/fibonacci-series-in-cpp/)canbeexecutedin many machines with littlebit or nochange.

1. **Mid-levelprogramminglanguage**

C++ has the ability to do both low-level & high-level programming. This is the reasonwhy C++ is known as a mid-level programming language. When we talk about low-level programming, C++ is used to develop system applications such as the kernel,driver,etc.

1. **Structuredprogramminglanguage**

In C++ programming, the code is modular with the help of functions, classes &objects, and the modules are loosely coupled. Modular code is easy to understand &modify.This makes C++astructuredprogramming language.

1. **RichLibrary**

Developers have access to lots of in-built functions provided by [C++ language.](https://www.edureka.co/blog/cpp-tutorial/) Thissaves time & makes development fast. Let’s look at some of the C++ header files &functionalitiesprovidedby it.

<iostream>:ContainsC++standardinput andoutput functions

<cmath>:Containsmath libraryfunctions

<cstdlib>:Containsfunctionforconversionsofnumberstotextandviseversa,memoryallocation, random numbersand variousotherutility functions.

<ctime>:Containsfunctionformanipulatingthetimeanddate

<fstream>: Contains function for functions that perform input from files on disk andoutputto files on disk

<memory>: Contains classes and functions used by the C++ Standard Library toallocatememory to theC++Standard Library containers

<iterator>:ContainsclassesforaccessingdataintheC++StandardLibrary containers

<algorithm>:ContainsfunctionsformanipulatingdatainC++StandardLibrarycontainers

1. **MemoryManagement**

C++ supports dynamic memory allocation. You can free the allocated memory at anytime.Not only thisC++also providesdynamicmemory managementtechniques.

1. **Powerful&Fast**

C++ is a fast language as compilation and execution time is less. Also, it has a widevarietyof datatypes, functions & operators.

1. **Pointers**

Pointers are variables that store the address of another variable. Pointer points to thememory location of a variable. C++ supports pointer and provides solutions to lots ofproblemsthat demand access to memory location.

1. **Compilerbased**

C++isacompiler-basedprogramminglanguage.Withoutcompilation,noC++program can be executed. The compiler first compiles the C++ program and then it isexecuted.

1. **C++Features:Syntaxbasedlanguage**

C++ is a language that complies strongly with syntax. Language following rules andregulations very strictly is known as tight syntax-based language. C, C++[, Java](https://www.edureka.co/blog/java-tutorial/), .netaresomeof theexamples.

**Algorithm:**

Step1:Start

Step2:CreateClass

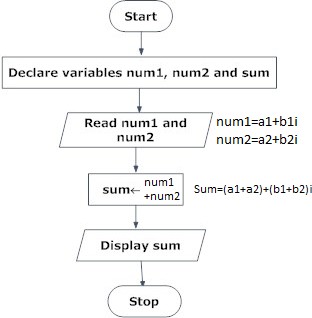
Step3:Definememberfunctions(Add,Sub,Mul)Step4:Createobjects

Step6:Entertwocomplexnumbers

Step7:AccessingofmemberfunctionbyobjectsStep8:Displaythevalues

Step9:Stop

**Flowchart:**



**CommandsUsedinProgram:**

1. **Cout<<**

Itisusedto produceoutputonthestandardoutput devicewhichisusuallythedisplayscreen.

1. **Cin>>**

**cin**statementistheinstanceoftheclassistreamandisusedtoreadinputfromthestandardinput devicewhichisusually a keyboard.

1. **Arithmetic’sOperators**

+(Addition),-(Subtraction),\*(Multiplication),/ (Division)

1. **Class**

A class in C++ is a user-defined type or data structure declared with keyword class that has dataand functions as its members whose access is governed by the three access specifiers private,protectedorpublic. Bydefaultaccessto membersofaC++ class is private.

1. **Setvalue()**

The **SetValue**() method sets a value to the element at the specified position in the one-dimensionalArray.

1. **Return0**

The**return**valueofthe**main**functionisconsideredthe"Exit Status" oftheapplication.

**Conclusion**:-

**Q&A**

Q1.WhatisC++? Whoinvented C++?

Q2.Whatiscasesensitivelanguage? IsC++ case sensitiveorinsensitive?

Q3.WhatisDatatype? Write C++onlytwoDataTypes.

Q4.WhatisVariable? What aretypesofvariables?

**Program**

#include<iostream>

using namespace std;

class complex

{

float x,y;

public:

complex()

{

x=0;

y=0;

}

complex operator+(complex);

complex operator\*(complex);

friend istream&operator >>(istream&input,complex&t)

{

cout<<"Enter the real part:";

input>>t.x;

cout<<"Enter the imaginary part:";

input>>t.y;

return input;

}

friend ostream&operator <<(ostream&output,complex&t)

{

output<<t.x<<"+"<<t.y<<"i\n";

return output;

}

};

complex complex::operator+(complex c)

{

complex temp;

temp.x=x+c.x;

temp.y=y+c.y;

return(temp);

}

complex complex::operator\*(complex c)

{

complex temp2;

temp2.x=(x\*c.x)-(y\*c.y);

temp2.y=(y\*c.x)+(x\*c.y);

return (temp2);

}

int main()

{

complex c1,c2,c3,c4;

cout<<"Default constructor value=\n";

cout<<c1;

cout<<"\nEnter the 1st number\n";

cin>>c1;

cout<<"\nEnter the 2nd number\n";

cin>>c2;

c3=c1+c2;

c4=c1\*c2;

cout<<"\nThe first number is ";

cout<<c1;

cout<<"\nThe second number is ";

cout<<c2;

cout<<"\nThe addition is ";

cout<<c3;

cout<<"\nThe multiplication is ";

cout<<c4;

return 0;

}

**Output**

Default constructor value=

0+0i

Enter the 1st number

Enter the real part:3

Enter the imaginary part:4

Enter the 2nd number

Enter the real part:4

Enter the imaginary part:7

The first number is 3+4i

The second number is 4+7i

The addition is 7+11i

The multiplication is -16+37i

**EXPERIMENTNO.02**

**TITLE:-StudentDatabase**

**ProblemStatement:**-DevelopaprograminC++tocreateadatabaseofstudentsinformationsystemcontainingthefollowinginformation:Name,Rollnumber,Class,Division, Date of Birth, Blood Group, Contact Address, Telephone Number, Driving Licensenumber. And other. Construct the database with suitable member functions. Make use ofconstructor, default constructor, copy constructor, destructor, static member function, friendclass, this pointer, inline code and dynamic memory allocation operators-new and delete aswellas exception handling.

**SoftwareUsed**:-DevC++

**Competency/Skills:**

Thestudents willbe abletoimprovethefollowing skills

1. To understand and apply the concepts like inheritance, polymorphism, exception handlingandgeneric structurefor implementingreusable programmingcodes.
2. To analyze the concepts of file and apply it while storing and retrieving the data fromsecondarystorages
3. Tolearnbasics,featuresand futureofC++programming

**LearningObjectives:**

Thestudent should beable to:

1. Tolearnproblemsolvingwithcomputers
2. Tounderstandanddevelop algorithmsandimplementprograms usingC++language.
3. TolearnfeaturesofObjectOrientedProgrammingusingC++programming.

**ExpectedOutcomes:**

Thestudentsshouldacquirethefollowingskill bythecompletion ofthisexperience:

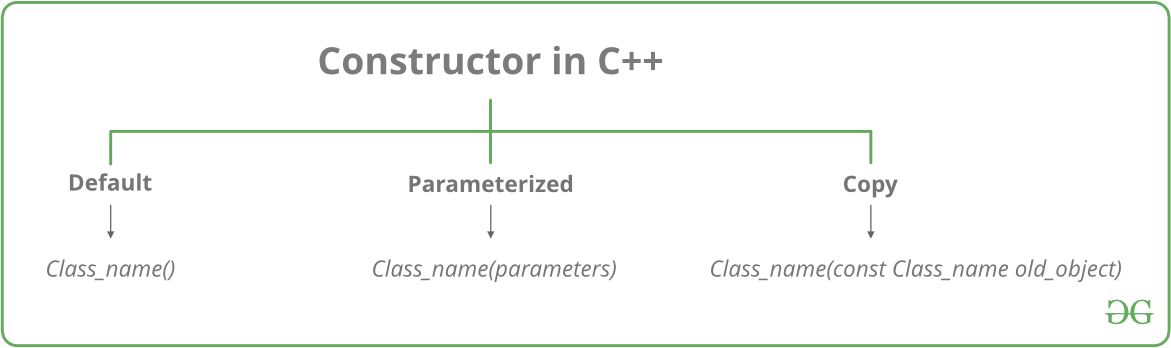
1. Inculcateandapplyvariousskillsin computergraphics
2. Exhibit the programming skills for the problems those require the writing of welldocumentedprogramsincludinguseofthelogicalconstructsoflanguage,C++.

**PracticalSignificance:**

Inobjectorientedprogrammingtherearevariousmethodsareusedtowriteprograms,usingthisexperiment theconceptsofstudentsdatabasecan beimplemented.

**Theory**:-

**Constructor**

Aconstructorisamemberfunctionofaclasswhichinitializesobjectsofaclass.InC++,Constructorisautomaticallycalledwhenobject(instanceofclass)create.Itisspecialmemberfunctionoftheclass.

* 1. [**Default Constructors:**](https://www.geeksforgeeks.org/c-internals-default-constructors-set-1/) Default constructor is the constructor which doesn’t takeanyargument.It has no parameters.
  2. **Parameterized Constructors:** It is possible to pass arguments to constructors.Typically,theseargumentshelpinitializeanobjectwhenitiscreated.Tocreateaparameterizedconstructor,simplyaddparameterstoitthewayyouwouldtoanyother function. When you define the constructor’s body, use the parameters toinitializetheobject.
  3. **Copy Constructor:** A copy constructor is a member function which initializesanobject using anotherobject of thesameclass.

**StaticMemberFunction**

By declaring a function member as static, you make itindependent of any particular objectof the class. A static member function can be called even if no objects of the class exist andthe **static** functionsareaccessedusingonlytheclassnameandthescoperesolutionoperator**::**.

A static member function can only access static data member, other static member functionsandany other functions from outsidethe class.

**Friend Class**

A **friend function** of a **class** is defined outside that **class**' scope but it has the right to access all private andprotectedmembers ofthe**class**. Eventhoughthe prototypesfor**friendfunctions**appearin

the**class**definition, **friends**are notmember**functions**.

**ThisPointer**

Every object in C++ has access to its own address through an important pointer called this pointer. The this pointer is an implicit parameter to all member functions. Therefore, inside a

member function, this may be used to refer to the invoking object Only member functions have a thispointer.

**Algorithm:**

Step1:Start

Step2:CreateClass

Step3:Definememberfunctions(Add,Sub,Mul)Step4:Createobjects

Step6:Entertwocomplexnumbers

Step7:AccessingofmemberfunctionbyobjectsStep8:Displaythevalues

Step9:Stop

**CommandsUsedinProgram:**

1. **String**

Inordertousethe**string**datatype,the **C++stringheader**<**string**>mustbeincludedatthetopof the program.

1. **Cstring**

Thisheaderfiledefinesseveralfunctionsto manipulate*Cstrings*and arrays.

1. **Cout<<**

It is used to produce output on the standard output device which is usually the displayscreen.

1. **Cin>>**

cin statement is the instance of the class istream and is used to read input from thestandardinput device which is usually a keyboard.

1. **Class**

A class in C++ is a user-defined type or data structure declared with keyword classthat has data and functions as its members whose access is governed by the threeaccess specifiers private, protected or public. By default access to members of a C++classis private.

1. **StaticVariable**

A static variable is a variable that is declared using the keyword static. The space forthe static variable is allocated only one time and this is used for the entirety of theprogram.Oncethis variableisdeclared, itexists tillthe programexecutes.

Syntax-staticdata\_typevariable\_name

1. **StaticFunction**

By declaring a function member as static, you make it independent of any particularobject of the class. A static member function can be called even if no objects of theclass exist and the static functions are accessed using only the class name and thescoperesolutionoperator::.

1. **Friend Class**

Afriendclasscanaccessprivateandprotectedmembersofotherclassinwhichitisdeclaredasfriend.Itissometimesusefultoallowaparticularclasstoaccessprivatemembersofotherclass.

1. **Friendfunction**

A friend function of a class is defined outside that class' scope but it has the right toaccessallprivateandprotectedmembersoftheclass.Eventhoughtheprototypesforfriendfunctions appear intheclassdefinition,friendsarenotmemberfunctions.

1. **PointerVariable**

A pointer is a variable that stores the address of another variable. Unlikeothervariables thatholdvaluesofacertaintype,pointerholdstheaddressofa variable. For example, an integer variable holds (or you can say stores) an integervalue,however an integer pointerholds the address ofainteger variable.

Syntax-data\_type\*variable\_name(pointerto avariable)

1. **Strcpy**

Thestrcpy()functioninC++copiesa characterstringfrom sourcetodestination.

Syntax-char\*strcpy(char\*dest, constchar\*src);

1. **Thispointer**

The**this**pointerholdstheaddressofcurrentobject,insimplewordsyoucansaythatthis[pointer](https://beginnersbook.com/2017/08/cpp-pointers/)points to thecurrent object of theclass.

**13.->**

the->(arrow)operatorareusedtoreferenceindividualmembersofclasses,structures,andunions.

(pointer\_name)->(variable\_name)

1. **Dowhile**

The do/while loop is a variant of the while loop. This loop will execute the code blockonce, before checking if the condition is true, then it will repeat the loop as long as theconditionis true.

do {

*//codeblockto beexecuted*

}

while(*condition*);

1. **Switch**

Switch case statementis used when wehavemultiple conditions and we needto perform different action based on the condition. When we have multipleconditions and we need to execute a block of statements when a particularconditionis satisfied.

1. **New**

The operator allocatesmemory toavariable.

new

Thereisfollowinggenericsyntaxtouse**new**operatortoallocatememorydynamicallyforany data-type.

Syntax-newdata-type;

1. **Break**

Thebreak statement canalso be usedto jump outof a loop.

Syntax-break;

1. **ScopeResolutionOperator::**

InC++,scoperesolutionoperatoris**::**.Itisusedforfollowingpurposes.

* 1. Toaccessaglobalvariablewhenthereisalocalvariablewithsamename
  2. Todefineafunctionoutsideaclass.

1. **cin.ignore()**

The cin.ignore() function is used which is used to ignore or clear one or more characters fromthe input buffer. A temporary storage area is called buffer.cin.ignore (120, '/ n'); the particularfunction skips the next input character 120 or to skip the characters until a new line characteris read.

1. **Return0**

The return valueofthe main functionisconsideredthe"ExitStatus"oftheapplication.

**Conclusion**:-

**Q&A**

Q1.Defineaccessspecifieranditsvarioustypesin C++

Q2.Explain‘this’pointer?

Q3.Whydo weneed the Friendclass andfunction?

Q4.Whatis adestructor?

**Program**

//student information

#include <iostream>

#include<string.h>

using namespace std;

class student

{

public:

int roll\_no;

char clas[10];

int sr\_no;

long int tele\_no;

char name[20];

char div;

char blood\_grp;

char DOB[10];

static int count;

void getdata();//function declear

friend void display(student & obj);//declear

student() //Constructor

{

roll\_no=0;

cout<<"\tConstructor";

roll\_no=count;

count++;

}

~student() //Destructor

{

cout<<"\nDestructor";

cout<<"\nDestroying the object";

count--;

}

student(int roll\_no)

{

this->roll\_no=roll\_no;

}

student (student & obj)

{

roll\_no=obj.roll\_no;

strcpy(name,obj.name);

strcpy(DOB,obj.DOB);

strcpy(clas,obj.clas);

blood\_grp=obj.blood\_grp;

div=obj.div;

tele\_no=obj.tele\_no;

sr\_no=count;

count++;

}

};

int student :: count=0;//intialize the out side of class static data member

void student::getdata()

{

cout<<"\n"<<"Enter the roll number of the student:";

cin>>roll\_no;

cout<<"\n"<<"Enter the name of the student:";

cin>>name;

cout<<"\n"<<"Enter the date of birth of the student:";

cin>>DOB;

cout<<"\n"<<"Enter the blood group of the student:";

cin>>blood\_grp;

cout<<"\n"<<"Enter the class of the student:";

cin>>clas;

cout<<"\n"<<"Enter the division of the student:";

cin>>div;

cout<<"\n"<<"Enter the contact of the student:";

cin>>tele\_no;

}

void display(student & obj)

{

cout<<"\n"<<obj.roll\_no;

cout<<"\t"<<obj.name;

cout<<"\t"<<obj.DOB;

cout<<"\t\t"<<obj.blood\_grp;

cout<<"\t"<<obj.clas;

cout<<"\t"<<obj.div;

cout<<"\t\t"<<obj.tele\_no;

}

int main()

{

student s1;

student s2(s1);

cout<<"\n Enter the details of a student:"<<"\n";

s1.getdata();

cout<<"All data is as displayed below:"<<"\n";

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

cout<<"\nROLL NO\tNAME\tDOB\t\tBLOOD GRP\tCLASS\tDIVISION\tCONTACT NO";

display(s1);

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

int i,n;

student \*s[50];//integer array of pointer

cout<<"\nEnter how many student object do you want us to create?"<<"\n";

cin>>n;

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

{

s[i]= new student();

}

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

{

s[i]->getdata();

}

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

{

display(\*s[i]);

}

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

{

delete (s[i]);

}

return 0;

}

**Output**

Constructor

Enter the details of a student:

Enter the roll number of the student:49

Enter the name of the student:Abdulrahman

Enter the date of birth of the student:25/09/2002

Enter the blood group of the student:O

Enter the class of the student:Computer

Enter the division of the student:B

Enter the contact of the student:123456789

All data is as displayed below:

---------------------------------------------------------------------

ROLLNO. NAME DOB BLOOD GRP CLASS DIVISION CONTACT NO

49 Abdulrahman 25/09/2002 O Computer B 7447885249

---------------------------------------------------------------------

Enter how many student object do you want us to create?

No

Constructor Constructor

Enter the roll number of the student:2

Enter the name of the student:ali

Enter the date of birth of the student:25/03/2002

Enter the blood group of the student:a

Enter the class of the student:civil

Enter the division of the student:b

Enter the contact of the student:123456789

Enter how many student object do you want us to create?

No

Destructor

Destroying the Object

Destructor

Destroying the Object

**EXPERIMENT NO. 03**

**TITLE:-PublishingCompany**

**Problem Statement:** - Imagine a publishing company which does marketing for book and audiocassette versions. Create a class publication that stores the title (a string) and price (type float) ofpublications. From this class derive two classes: book which adds a page count (type int) and tapewhichadds aplaying time in minutes(type float).

Write a program that instantiates the book and tape class, allows user to enter data and displays thedatamembers.Ifan exceptioniscaught,replaceallthedatamember values with zerovalues

**SoftwareUsed**:-DevC++

**Competency/Skills:**

Thestudents willbe abletoimprovethefollowing skills

1. To understand and apply the concepts like inheritance, polymorphism, exception handlingandgeneric structurefor implementingreusable programmingcodes.
2. To analyze the concepts of file and apply it while storing and retrieving the data fromsecondarystorages
3. Tolearnbasics,featuresand futureofC++programming

**LearningObjectives:**

Thestudent should beable to:

1. Tolearnproblemsolvingwithcomputers
2. Tounderstandand developalgorithms andimplement programsusingC++language.
3. TolearnfeaturesofObjectOrientedProgrammingusingC++programming.

**ExpectedOutcomes:**

Thestudentsshouldacquirethefollowingskill bythecompletion ofthisexperience:

1. Inculcateandapplyvariousskillsincomputergraphics
2. Exhibit the programming skills for the problems those require the writing of welldocumentedprogramsincludinguseofthelogicalconstructsoflanguage,C++.

**PracticalSignificance:**

Inobjectorientedprogrammingtherearevariousmethodsareusedtowriteprograms,usingthisexperiment theconceptsofpublishing companydatabasecan beimplemented.

**Theory**:-

DataType

Meaning

Size(inBytes)

int

Integer

2or4

Floating-point

4

DoubleFloating-point

8

Character

1

WideCharacter

2

Boolean

1

Empty

0

void

bool

wchar\_t

char

double

float

**CommandsUsedinProgram:**

1. **String**

In order to use the **string** data type, the **C++ string header** <**string**> must be included at thetopof the program.

1. **Conio.h**

The **conio stands for Console-Input-Output**. The conio.h is a non-standard header fileusedin C and C++ [programming](https://techsupportwhale.com/programming/).

1. **Cout<<**

It is used to produce output on the standard output device which is usually the displayscreen.

1. **Cin>>**

cin statement is the instance of the class istream and is used to read input from thestandardinput device which is usually a keyboard.

1. **Class**

A class in C++ is a user-defined type or data structure declared with keyword classthat has data and functions as its members whose access is governed by the threeaccess specifiers private, protected or public. By default access to members of a C++classis private.

1. **SubClass**

Wecanextendasubclass(orderivedclass)fromasuperclass(orbaseclass). Thesubclass*inherits*the members ofthesuperclass, known as*inheritance*.

Thesyntax forderiving asubclassfromasuperclass isasfollows:

**class SubclassName : inheritance-access-specifier SuperclassName**

**{**

**......**

**};**

1. **ScopeResolutionOperator::**

InC++,scoperesolutionoperatoris**::**.Itisusedforfollowingpurposes.

* 1. Toaccessaglobalvariablewhenthereisalocalvariablewithsamename
  2. Todefineafunctionoutsideaclass.

1. **Return0**

Thereturnvalueofthemain functionisconsideredthe"ExitStatus"oftheapplication.

**Conclusion**:-

**Q&A**

Q1.Defineinheritance.

Q2.Definetypes of class.

Q3.Why we areusescoperesolution operator?

Q4.Writesignificanceof“return0”.

**Program**

#include<iostream>

#include<string>

using namespace std;

//base class publication

class publication

{

private:

string title;

float prices;

public:

publication()//default condturctor

{

title="";

prices=0.0;

}

void get\_data()

{

cout<<"\nEnterTitle :"<<"\n";

cin.ignore();//clear input buffer

getline(cin,title);

cout<<"\nEnterPrice : "<<"\n";

cin>>prices;

}

void put\_data()

{

cout<<"\n \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \n";

cout<<"\n Information : " <<endl;

cout<<"\n Title :"<<title;

cout<<"\n Price :"<<prices;

}

};

class book: public publication

{

private:

int pages;

public:

book()//default condturctor

{

pages=0;

}

void get\_data()

{

publication::get\_data();

cout<<endl;

cout<<"Enter Page Count : \n";

cin>>pages;

}

void put\_data()

{

publication::put\_data();

try{

if(pages<0)

throw pages;}

catch(int f)

{

cout<<"\n error: pages not valid :"<<f;

pages=0;

}

cout<<"\n Pages Are :"<<pages<<"\n";

}

};

class tape: public publication

{

private:

float playtime;

public:

tape()

{

playtime=0.0;

}

void get\_data()

{

publication::get\_data();

cout<<"Enter Play Time Of Cassette \n";

cin>>playtime;

}

void put\_data()

{

publication::put\_data();

try

{

if(playtime<0.0)

throw playtime;

}

catch(float r)

{

cout<<"\n Error: Invalid Playtime : "<<playtime;

playtime=0.0;

}

cout<<"\n Playtime is : "<<playtime<<"\n";

}

};

int main()//main program

{

book b[10];// arrray of objects

tape t[10];

int choice=0,bookCount=0,tapeCount=0;

cout<<"-----------------------";

do

{

cout<<"\n 1. Add Book ";

cout<<"\n 2. Add Tape: ";

cout<<"\n 3. Display Book ";

cout<<"\n 4. Display Tape";

cout<<"\n 5. Exit:"<<endl;

cout<<"\n Enter Choice : ";

cin>>choice;

switch(choice)

{

case 1:

{

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

cout<<"Add Book: \n";

b[bookCount].get\_data();

bookCount++;

break;

}

case 2:

{

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

cout<<"Add Tape: \n";

t[tapeCount].get\_data();

tapeCount++;

break;

}

case 3:

{

cout<<"\n (books)";

for(int j=0;j<bookCount;j++)

{

b[j].put\_data();

}

break;

}

case 4:

{

cout<<"\n (tape)";

for(int j=0;j<tapeCount;j++)

{

t[j].put\_data();

}

break;

}

case 5:

{

cout<<"\*\*\*\*\*\*\*\*\*\*Program Execuited Successfully\*\*\*\*\*\*\*\*\*\*"<<endl;

exit(0);

}

default:

{

cout<<"\n Invalid";

}

}

}

while(choice!=5);

return 0;

}

**Output**

1. Add Book

2. Add Tape:

3. Display Book

4. Display Tape

5. Exit:

Enter Choice : 1

--------------

Add Book:

Enter Title :

c++

Enter Price :

200

Enter Page Count :

500

1. Add Book

2. Add Tape:

3. Display Book

4. Display Tape

5. Exit:

Enter Choice : 2

--------------

Add Tape:

Enter Title :

functions

Enter Price :

200

Enter Play Time Of Cassette

3

1. Add Book

2. Add Tape:

3. Display Book

4. Display Tape

5. Exit:

Enter Choice : 3

(books)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Information :

Title :c++

Price :200

Pages Are :500

1. Add Book

2. Add Tape:

3. Display Book

4. Display Tape

5. Exit:

Enter Choice : 4

(tape)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Information :

Title :functions

Price :200

Playtime is : 3

1. Add Book

2. Add Tape:

3. Display Book

4. Display Tape

5. Exit:

Enter Choice : 5

\*\*\*\*\*\*\*\*\*\*Program Execuited Successfully\*\*\*\*\*\*\*\*\*\*

**EXPERIMENTNO.04**

**TITLE:-Fileoperations**

**ProblemStatement:**-WriteaC++programthatcreatesanoutputfile,writesinformationtoit,closesthefile,openit againas aninputfileand readthe information fromthe file.

**SoftwareUsed**:-DevC++

**Competency/Skills:**

Thestudents willbe abletoimprovethefollowing skills

1. To understand and apply the concepts like inheritance, polymorphism, exception handlingandgenericstructureforimplementing reusableprogrammingcodes.
2. To analyze the concepts of file and apply it while storing and retrieving the data fromsecondarystorages
3. Tolearnbasics,featuresand futureofC++programming

**LearningObjectives:**

Thestudent should beable to:

1. Tolearnproblemsolvingwithcomputers
2. Tounderstandand developalgorithmsand implementprogramsusingC++language.
3. TolearnfeaturesofObjectOrientedProgrammingusingC++programming.

**ExpectedOutcomes:**

Thestudentsshould acquirethe followingskillby thecompletionofthisexperience:

1. Inculcateandapplyvariousskillsincomputergraphics
2. Exhibit the programming skills for the problems those require the writing of welldocumentedprogramsincludinguseofthelogicalconstructsoflanguage,C++.

**PracticalSignificance:**

Inobjectorientedprogrammingtherearevariousmethodsareusedtowriteprograms,usingthisexperiment theconceptsofpublishing companydatabasecan beimplemented.

**Theory**:-

File represents storage medium for storing data or information. Streams refer to sequence ofbytes. In Files we store data i.e. text or binary data permanently and use these data to read orwrite in the form of input output operations by transferring bytes of data. So we use the termFileStreams/Filehandling. Weusetheheader file<fstream>

* **ofstream:**ItrepresentsoutputStreamandthisisusedforwritinginfiles.
* **ifstream:**ItrepresentsinputStreamandthisisusedforreadingfromfiles.
* **fstream:** It represents both output Stream and input Stream. So it can read from filesandwriteto files.

OperationsinFileHandling:

* Creatingafile:open()
* Readingdata:read()
* Writingnewdata:write()
* Closingafile:close()

**CommandsUsedinProgram:**

1. **fstream**

In order to use the **string** data type, the **C++ string header** <**string**> must be included at thetopof the program.

1. **Conio.h**

The **conio stands for Console-Input-Output**. The conio.h is a non-standard header fileusedin C and C++ [programming](https://techsupportwhale.com/programming/).

1. **Cout<<**

It is used to produce output on the standard output device which is usually the displayscreen.

1. **Cin>>**

cin statement is the instance of the class istream and is used to read input from thestandardinput device which is usually a keyboard.

1. **Open**

An open file is represented within a program by a *stream* (i.e., an object of one ofthese classes;inthepreviousexample,thiswas myfile) andanyinputor outputoperationperformedonthisstreamobjectwillbeappliedtothephysicalfileassociatedto it.

**Syntax**-open(filename,mode);

Where filename is a string representing the name of the file to be opened, and mode isanoptional parameter with a combinationofthe following flags:

|  |  |
| --- | --- |
| ios::in | Openforinputoperations. |
| ios::out | Openforoutputoperations. |
| ios::binary | Openinbinarymode. |
| ios::ate | Settheinitialpositionattheendofthe file.  Ifthisflag isnot set,theinitialposition isthe beginningofthefile. |
| ios::app | Alloutputoperationsareperformedattheendofthefile,appendingthe content tothecurrent content of thefile. |
| ios::trunc | Ifthefileis openedforoutput operationsanditalready existed,itsprevious |
|  | contentisdeletedandreplacedbythenewone. |

1. **Close**

When we are finished with our input and output operations on a file we shall close itso that the operating system is notified and its resources become available again. Forthat, we call the stream's member function close. This member function takes flushestheassociated buffersand closes the file:

**Syntax-**myfile.close();

1. **ScopeResolutionOperator::**

InC++,scoperesolutionoperatoris**::**.Itisusedforfollowingpurposes.

* 1. Toaccessaglobalvariablewhenthereisalocalvariablewithsamename
  2. Todefineafunctionoutsideaclass.

1. **CheckingStateFlag**

The following member functions exist to check for specific states of a stream (all ofthemreturn aboolvalue):

**bad()**

Returns true if a reading or writing operation fails. For example, in the case that wetry to write to a file that is not open for writing or if the device where we try to writehasno spaceleft.

**fail()**

Returns true in the same cases as bad(), but also in the case that a format errorhappens,likewhen analphabeticalcharacterisextractedwhenwearetryingtoreadaninteger number.

**eof()**

Returnstrueif afileopen forreading has reachedthe end.

**good()**

It is the most generic state flag: it returns false in the same cases in which calling anyof the previous functions would return true. Note that good and bad are not exactopposites(good checksmorestateflagsat once).

1. **Return0**

Thereturnvalueofthemain functionisconsideredthe"ExitStatus"oftheapplication.

**Conclusion**:-

**Q&A**

Q1.Define file.

Q2.Definefileoperations.

Q3.Whichcommandisused toclosethefile?

**Program**

#include <fstream>

#include <iostream>

using namespace std;

int main ()

{

char data[100];

// open a file in write mode.

ofstreamoutfile;

outfile.open("afile.txt");

cout<< "Writing to the file" <<endl;

cout<< "Enter your name: ";

cin.getline(data, 100);

// write inputted data into the file.

outfile<< data <<endl;

cout<< "Enter your age: ";

cin>> data;

cin.ignore();

// again write inputted data into the file.

outfile<< data <<endl;

// close the opened file.

outfile.close();

// open a file in read mode.

ifstreaminfile;

infile.open("afile.txt");

cout<< "Reading from the file" <<endl;

infile>> data;

// write the data at the screen.

cout<< data <<endl;

// again read the data from the file and display it.

infile>> data;

cout<< data <<endl;

// close the opened file.

infile.close();

return 0;

}

**Output**

Writing to the file

Enter your name: Abdulrahman

Enter your age: 20

Reading from the file

Abdulrahman

20

**EXPERIMENT NO.05**

**TITLE:-Sortingofarray**

**ProblemStatement:**-Writeafunctiontemplateforselectionsortthatinputs,sortsandoutputsanintegerarrayand afloatarray.

**SoftwareUsed**:-DevC++

**Competency/Skills:**

Thestudents willbe abletoimprovethefollowing skills

1. To understand and apply the concepts like inheritance, polymorphism, exception handlingandgeneric structurefor implementingreusable programmingcodes.
2. To analyze the concepts of file and apply it while storing and retrieving the data fromsecondarystorages
3. Tolearnbasics,featuresand futureofC++programming

**LearningObjectives:**

Thestudent should beable to:

1. Tolearnproblemsolvingwithcomputers
2. Tounderstandand developalgorithms andimplement programsusingC++language.
3. TolearnfeaturesofObjectOrientedProgrammingusingC++programming.

**ExpectedOutcomes:**

Thestudentsshouldacquirethefollowingskill bythecompletion ofthisexperience:

1. Inculcateandapplyvariousskillsincomputergraphics
2. Exhibit the programming skills for the problems those require the writing of welldocumentedprogramsincludinguseofthelogicalconstructsoflanguage,C++.

**PracticalSignificance:**

Inobjectorientedprogrammingtherearevariousmethodsareusedtowriteprograms,usingthisexperiment theconceptsofpublishing companydatabasecan beimplemented.

**Theory**:-

Theselectionsortalgorithmsortsanarraybyrepeatedlyfindingtheminimumelement(considering ascending order) from unsorted part and putting it at the beginning. Thealgorithmmaintainstwosubarraysinagivenarray.

1. Thesubarraywhichisalreadysorted.
2. Remainingsubarraywhichisunsorted.

Ineveryiterationofselectionsort,theminimumelement(consideringascendingorder)fromtheunsortedsubarrayispickedandmovedtothesortedsubarray.

**CommandsUsedinProgram:**

* 1. **iostream**
  2. **Cout<<**

Itisusedtoproduceoutputonthestandardoutputdevicewhichisusuallythedisplayscreen.

* 1. **Cin>>**

cinstatementistheinstanceoftheclassistreamandisusedtoreadinputfromthestandardinput device which is usually a keyboard.

* 1. **Function**

Afunctionisablockofcodewhichonlyrunswhenitiscalled.Youcanpassdata,knownasparameters,intoa function.

Functions are used to perform certain actions, and they areimportant for reusing code: Define the code once, and use it manytimes.

**Syntax-**

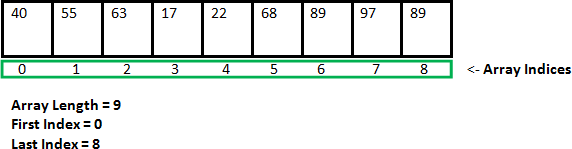
void*myFunction*(){

//codetobe executed

}

* 1. **Array**

An **array** in C or **C++** is a collection of items stored at contiguous memory locationsand elements can be accessed randomly using indices of an **array**. ... They can beused to store collection of primitive data types such as int, float, double, char, etc ofany particulartype.



**Syntax**

Data\_typearrayName[arraySize];

* 1. **For**

A**for**loopisarepetitioncontrolstructurethatallowsyoutoefficientlywritealoopthatneedsto executeaspecificnumberoftimes.

Syntax- for(init;condition;increment){

statement(s);

}

* 1. **Return0**

Thereturnvalueofthemain functionisconsideredthe"ExitStatus"oftheapplication.

**Conclusion**:-

**Q&A**

Q1.Definefunction.

Q2.Definetypes ofloop.

Q3.What isthe differencebetweenintegerand floatdatatype?

**Program**

#include<iostream>

#define size 10

using namespace std;

int n;

template<class T>

void selection(T A[size])

{

int i,j,min;

T temp;

for(i=0;i<n-1;i++)

{

min=i;

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

{

if(A[j]<A[min])

min=j;

}

temp=A[i];

A[i]=A[min];

A[min]=temp;

}

cout<<"\nSorted Array In Ascending Order: ";

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

{

cout<<" "<<A[i];

}

cout<<"\nSorted Array In Descending Order : ";

for(i=n-1;i>=0;i--)

{

cout<<" "<<A[i];

}

}

int main()

{

int choice;

char C[size];

int A[size];

float B[size];

int i;

cout<<"-----------------------";

do

{

cout<<"\n 1. Integer : ";

cout<<"\n 2. Float : ";

cout<<"\n 3. Exit : "<<endl;

cout<<"\n Enter Choice : ";

cin>>choice;

switch(choice)

{

case 1:

{

cout<<"\nEnter Total Number Of Integer Elements:";

cin>>n;

cout<<"\nEnter Integer Elements:";

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

{

cin>>A[i];

}

selection(A);

break;

}

case 2:

{

cout<<"\nEnter Total Number Of Float Elements:";

cin>>n;

cout<<"\nEnter Float Elememts:";

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

{

cin>>B[i];

}

selection(B);

break;

}

case 3:

{

cout<<"\*\*\*\*\*\*\*\*\*\*Program Exited Successfully\*\*\*\*\*\*\*\*\*\*"<<endl;

exit(0);

}

default:

{

cout<<"\n Invalid";

}

}

}

while(choice!=4);

return 0;

}

**Output**

-----------------------

1. Integer :

2. Float :

3. Exit :

Enter Choice : 1

Enter Total Number Of Integer Elements:3

Enter Integer Elements:2

3

5

Sorted Array In Ascending Order: 2 3 5

Sorted Array In Descending Order : 5 3 2

1. Integer :

2. Float :

3. Exit :

Enter Choice : 2

Enter Total Number Of Float Elements:3

Enter Float Elememts:7.7

7.8

8.9

Sorted Array In Ascending Order: 7.7 7.8 8.9

Sorted Array In Descending Order : 8.9 7.8 7.7

1. Integer :

2. Float :

3. Exit :

Enter Choice : 3

\*\*\*\*\*\*\*\*\*\*Program Exited Successfully\*\*\*\*\*\*\*\*\*\*

**EXPERIMENTNO.06**

**TITLE:-PublishingCompany**

**ProblemStatement:**-WriteC++programusingSTLforsorting&searchinguserdefinedrecordssuchaspersonalrecords (Name, DOB, Telephonenumberetc.)usingvectorcontainer.

OR

WriteC++programusingSTLforsorting&searchinguserdefinedrecordssuchaspersonalrecords (Item code, name, cost,quantity etc.) using vectorcontainer.

**SoftwareUsed**:-DevC++

**Competency/Skills:**

Thestudents willbe abletoimprovethefollowing skills

1. To understand and apply the concepts like inheritance, polymorphism, exceptionhandlingandgenericstructureforimplementingreusable programmingcodes.
2. To analyze the concepts of file and apply it while storing and retrieving the data fromsecondarystorages
3. Tolearnbasics,featuresand futureofC++programming

**LearningObjectives:**

Thestudent should beable to:

1. Tolearnproblemsolvingwithcomputers
2. Tounderstandand developalgorithmsand implementprogramsusingC++language.
3. TolearnfeaturesofObjectOrientedProgrammingusingC++programming.

**ExpectedOutcomes:**

Thestudentsshouldacquirethefollowingskill bythecompletion ofthisexperience:

1. Inculcateandapplyvariousskillsincomputergraphics
2. Exhibit the programming skills for the problems those require the writing of welldocumentedprograms includinguseofthelogicalconstructsof language,C++.

**PracticalSignificance:**

Inobjectorientedprogrammingtherearevariousmethodsareusedtowriteprograms,usingthisexperiment theconceptsofpublishing companydatabasecan beimplemented.

**Theory**:-

The C++ STL (Standard Template Library) is a powerful set of C++ template classes toprovide general-purpose classes and functions with templates that implement many popularandcommonlyused algorithmsand data structureslikevectors,lists, queues,andstacks.

Atthecore oftheC++StandardTemplateLibraryarefollowingthreewell-structuredcomponents −

|  |  |
| --- | --- |
| **Sr.No** | **Component&Description** |
| 1 | **Containers**  Containersareusedtomanagecollectionsofobjectsofacertainkind.Thereareseveraldifferenttypes of containers like deque,list, vector, map etc. |
| 2 | **Algorithms**  Algorithmsactoncontainers.Theyprovidethemeansbywhichyouwillperforminitialization,sorting, searching,and transformingofthe contentsof containers. |
| 3 | **Iterators**  Iteratorsareusedtostepthroughtheelementsofcollectionsofobjects.Thesecollectionsmaybecontainers orsubsets of containers. |

**CommandsUsedinProgram:**

1. **Iostream**

In the C++ programming language, input/output library refers to a family of classtemplatesandsupportingfunctionsintheC++StandardLibrarythatimplementstream-basedinput/output capabilities.

1. **Algorithm**

The header <algorithm> defines a collection of functions especially designed to beused on ranges of elements.A range is any sequence of objects that can be accessed through iterators or pointers,such as an array or an instance of some of the [STL containers](http://www.cplusplus.com/stl). Notice though, thatalgorithmsoperate through iterators directly on the values, notaffecting inany waythe structure of any possible container (it never affects the size or storage allocation ofthecontainer).

1. **Vector**

Vectors in C++ are sequence containers representing arrays that can change in size.They use contiguous storage locations for their elements, which means that theirelementscanalsobeaccessedusingoffsetsonregularpointerstoitselements,andjustas efficiently as in arrays.

Vectors in C++ are thedynamic arrays that are used to store data. Unlike arrays,which are used to store sequential data and are static in nature, Vectors provide moreflexibilityto the program.

Syntax-vector<object\_type>variable\_name;

1. **Booloperator**

A boolean datatypein **C++** isdefinedusingthekeyword **bool** ....Althoughanynumerical value can be assigned to a **boolean**variable in C++, all values other than 0areconsidered to betrue and stored as1, while0 isconsidered to befalse.

1. **Const**

Theconst keywordspecifiesthatavariable'svalueisconstantandtellsthecompilertoprevent theprogrammer from modifying it.

**6.&**

If you use & in the left-hand side of a variable declaration, it means that you expect tohave a [reference](https://www.tutorialspoint.com/cplusplus/cpp_references.htm)to the declared type. It can be used in any type of declarations (localvariables,class members,method parameters).

1. **Switch**

Switch case statement is used when we have multiple conditions and we need toperformdifferent actionbased on the condition.

1. **Begin**

begin() function is used to return an iterator pointing to the first element of the vectorcontainer.

1. **End**

end()functionisusedtoreturnaniteratorpointingtonexttolastelementofthevectorcontainer.

1. **Dowhile**

The do/while loop is a variant of the while loop. This loop will execute the code blockonce, before checking if the condition is true, then it will repeat the loop as long as theconditionis true.

1. **Iterator**

Iterators are used to point at the memoryaddresses of STL containers. They areprimarily used in sequence of numbers, characters etc. They reduce the complexityandexecution time of program.

Iterators are used to traverse from one element to another element, a process is knownas **iterating** through the container. The main advantage of an **iterator** is to provide acommoninterfaceforallthecontainerstype. **Iterators** makethealgorithmindependentof the typeof thecontainer used.

An*iterator* isanyobjectthat,pointingtosomeelementinarangeofelements(suchas an array or a [container](https://www.cplusplus.com/stl)), has the ability to iterate through the elements of that rangeusingasetofoperators(withatleasttheincrement(++)anddereference(\*)operators).

1. **Cout<<**

It is used to produce output on the standard output device which is usually the displayscreen.

1. **Cin>>**

cin statement is the instance of the class istream and is used to read input from thestandardinput device which is usually a keyboard.

1. **Class**

A class in C++ is a user-defined type or data structure declared with keyword classthathasdataandfunctionsasitsmemberswhoseaccessisgovernedbythethree

accessspecifiersprivate,protectedorpublic.BydefaultaccesstomembersofaC++classis private.

1. **Return0**

Thereturnvalueofthemain functionisconsideredthe"ExitStatus"oftheapplication.

**Conclusion**:-

**Q&A**

Q1.DefineSTL.

Q2.Definetemplate.

**Program**

#include <iostream> //standard input output stream header file

#include <algorithm> //The STL algorithms are generic because they can operate on a variety of data structures

#include <vector> //The header file for the STL vector library is vector.

using namespace std;

class Item

{

public:

char name[10];

int quantity;

int cost;

int code;

bool operator==(const Item& i1) //Boolean operators allow you to create more complex conditional statements

{

if(code==i1.code) //operator will return 1 if the comparison is true, or 0 if the comparison is false

return 1;

return 0;

}

bool operator<(const Item& i1)

{

if(code<i1.code) //operator will return 1 if the comparison is true, or 0 if the comparison is false

return 1;

return 0;

}

};

vector<Item> o1;

void print(Item &i1);

void display();

void insert();

void search();

void dlt();

bool compare(const Item &i1, const Item &i2)

{

//if (i1.name != i2.name) return i1.cost < i2.cost;

return i1.cost< i2.cost;

}

int main()

{

int ch;

do

{

cout<<"\n\*\*\*\*\* Menu \*\*\*\*\*";

cout<<"\n1.Insert";

cout<<"\n2.Display";

cout<<"\n3.Search";

cout<<"\n4.Sort";

cout<<"\n5.Delete";

cout<<"\n6.Exit";

cout<<"\nEnter your choice:";

cin>>ch;

switch(ch)

{

case 1:

insert();

break;

case 2:

display();

break;

case 3:

search();

break;

case 4:

sort(o1.begin(),o1.end(),compare);

cout<<"\n\n Sorted on Cost";

display();

break;

case 5:

dlt();

break;

case 6:

exit(0);

}

}while(ch!=7);

return 0;

}

void insert()

{

Item i1;

cout<<"\nEnter Item Name:";

cin>>i1.name;

cout<<"\nEnter Item Quantity:";

cin>>i1.quantity;

cout<<"\nEnter Item Cost:";

cin>>i1.cost;

cout<<"\nEnter Item Code:";

cin>>i1.code;

o1.push\_back(i1);

}

void display()

{

for\_each(o1.begin(),o1.end(),print);

}

void print(Item &i1)

{

cout<<"\n";

cout<<"\nItem Name:"<<i1.name;

cout<<"\nItem Quantity:"<<i1.quantity;

cout<<"\nItem Cost:"<<i1.cost;

cout<<"\nItem Code:"<<i1.code;

}

void search()

{

vector<Item>::iterator p;

Item i1;

cout<<"\nEnter Item Code to search:";

cin>>i1.code;

p=find(o1.begin(),o1.end(),i1);

if(p==o1.end())

{

cout<<"\nNot found.";

}

else

{

cout<<"\nFound.";

}

}

void dlt()

{

vector<Item>::iterator p;

Item i1;

cout<<"\nEnter Item Code to delete:";

cin>>i1.code;

p=find(o1.begin(),o1.end(),i1);

if(p==o1.end())

{

cout<<"\nNot found.";

}

else

{

o1.erase(p);

cout<<"\nDeleted.";

}

}

**Output**

\*\*\*\*\* Menu \*\*\*\*\*

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:1

Enter Item Name:book

Enter Item Quantity:3

Enter Item Cost:200

Enter Item Code:1234

\*\*\*\*\* Menu \*\*\*\*\*

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:2

Item Name:book

Item Quantity:3

Item Cost:200

Item Code:1234

\*\*\*\*\* Menu \*\*\*\*\*

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:3

Enter Item Code to search:1234

Found.

\*\*\*\*\* Menu \*\*\*\*\*

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:4

Sorted on Cost

Item Name:book

Item Quantity:3

Item Cost:200

Item Code:1234

\*\*\*\*\* Menu \*\*\*\*\*

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:5

Enter Item Code to delete:1234

Deleted.

\*\*\*\*\* Menu \*\*\*\*\*

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:6

**EXPERIMENTNO.07**

**TITLE:-Populationsofthestates**

**Problem Statement:** - Write a program in C++ to use map associative container. The keys will bethe names of states and the values will be the populations of states. When the program runs, the user isprompted to typenameof astate.Theprogram then looksin the map, using thestatename as anindexandreturns the population ofthestate.

**SoftwareUsed**:-DevC++

**Competency/Skills:**

Thestudents willbe abletoimprovethefollowing skills

1. To understand and apply the concepts like inheritance, polymorphism, exception handlingandgeneric structurefor implementingreusable programmingcodes.
2. To analyze the concepts of file and apply it while storing and retrieving the data fromsecondarystorages
3. Tolearnbasics,featuresand futureofC++programming

**LearningObjectives:**

Thestudent should beable to:

1. Tolearnproblemsolvingwithcomputers
2. Tounderstandand developalgorithmsand implementprogramsusingC++language.
3. TolearnfeaturesofObjectOrientedProgrammingusingC++programming.

**ExpectedOutcomes:**

Thestudentsshouldacquirethefollowingskill bythecompletion ofthisexperience:

1. Inculcateandapplyvariousskillsincomputergraphics
2. Exhibit the programming skills for the problems those require the writing of welldocumentedprogramsincludinguseofthelogicalconstructsoflanguage,C++.

**PracticalSignificance:**

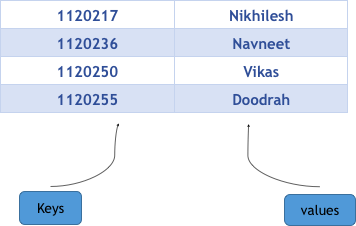
Inobjectorientedprogrammingtherearevariousmethodsareusedtowriteprograms,usingthisexperiment theconceptsofpublishing companydatabasecan beimplemented.

**Theory**:-

**Maps** are used to replicate associative arrays. Maps contain sorted **key-value** pair,inwhicheachkeyisuniqueandcannotbechanged,andit canbeinsertedor

deleted but cannot be altered. Value associated with keys can be altered. We cansearch,removeandinsert in amap withinO(n)timecomplexity.

For example: A map of students where **roll number** is the key and **name** is thevalue can be represented graphicallyas:



Noticethatkeysare arrangedinascendingorder,itsbecausemapsalwaysarrangeitskeysinsortedorder.Incasethe keysareofstringtype,theyaresorted lexicographically.

Mapscaneasily becreatedusingthefollowingstatement:

map<key\_type,value\_type>map\_name

This will create a map with key of type **Key\_type**and value of type **value\_type**. One thingwhich is to remembered is that key of a map and corresponding values are always insertedasapair,you cannotinsert onlykeyorjusta value in amap.

**CommandsUsedinProgram:**

1. **String**

Inordertousethe**string**datatype,the **C++stringheader**<**string**>mustbeincludedatthetopof the program.

1. **Conio.h**

The **coniostandsforConsole-Input-Output**.Theconio.hisanon-standardheaderfileusedin C and C++ [programming](https://techsupportwhale.com/programming/).

1. **Cout<<**

Itisusedtoproduceoutputonthestandardoutputdevicewhichisusuallythedisplayscreen.

1. **Cin>>**

cinstatementistheinstanceoftheclassistreamandisusedtoreadinputfromthestandardinput device which is usually a keyboard.

1. **ScopeResolutionOperator::**

InC++,scoperesolutionoperatoris**::**.Itisusedforfollowingpurposes.

* 1. Toaccessaglobalvariablewhenthereisalocalvariablewithsamename
  2. Todefineafunctionoutsideaclass.

1. **Return0**

Thereturnvalueofthemain functionisconsideredthe"ExitStatus"oftheapplication.

**Conclusion**:-

**Q&A**

Q1.Definemapinc++.

Q2.Howto declaremap.

**Program**

#include<iostream>

#include<map>

#include<string>

using namespace std;

int main()

{

map<string,int>first;

//initializing

first["Maharashtra"]= 123144223;

first["Gujarat"]= 63872399 ;

first["Rajasthan"]= 81032689 ;

first["Bihar"]= 124799926;

first["Kashmir"]= 13606320;

first["Punjab"]= 30141373 ;

first["UP"]= 30141373 ;

//Alternative mathod

map<string,int>::iterator it;

cout<<"==========Population of states in India============\n\n";

cout<<"\n Size of populationMap=\t"<<first.size()<<"\n"<<"\n";

for(it=first.begin();it!=first.end();it++)

{

cout<<"Name of State \t"<<it->first<<"=>"<<"Population\t"<<it->second<<'\n' ;

}

string c;

cout<<" \n Write State Name \t";

cin>>c;

cout<<" Population Of \t"<< c <<" Is \t"<<first[c];

return 0;

}

**Output**

==========Population of states in India============

Size of populationMap= 7

Name of State Bihar=>Population 124799926

Name of State Gujarat=>Population 63872399

Name of State Kashmir=>Population 13606320

Name of State Maharashtra=>Population 123144223

Name of State Punjab=>Population 30141373

Name of State Rajasthan=>Population 81032689

Name of State UP=>Population 30141373

Write State Name

Maharashtra

Population Of Maharashtra Is 123144223