DOMS | Page No.
Date / /

Exp-1

Q1) Write a program to declare class students having data members as roll no & name. Accept & display data for one object.

Q2) Write a program to declare a class book having data members as book name, price & no. Accept data for two objects and display the name of book having greater price.

Q3) Write a program to declare a class time. Accept the time in (HH:MM:SS) in this format and convert it into total seconds and display

---

q1) Add 2 no's.
2) Arithmatic operation (+, -, /, *) using switch.
3) Check even or odd.
4) Print 1 to 10 no's using for loop.
5) Print 1 to 10 using while loop.
6) Print.

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

b)
```
1
12
123
1234
12345
```

c)
```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

DOMS | Page No.
Date / /

1 Write C++ program to add 2 nos.

```
#include <iostream>
using namespace std;

int main()
{
  int a, b, sum;
  cout << "Enter 2 nos = ";
  cin >> a >> b
  sum = a+b;
  cout << "sum of 2 no's is " << sum;
  return 0;
}
```

out <<
in >>

Q Arithmatical Operations using switch.

```
#include <iostream>
using namespace std;

int main()
{
  int choice a, b;
  cout <<"enter 2 no's = ";
   cin >> a >>b;
  cout << " 1. Addition \n 2. substraction \n 3. Multiplication
          \n 4. Division";
   cin >> choice;

switch (choice)
 {
 case 1:
      cout <<a+b;
          break;
Case 2: 
      cout <<a-b;
          break;
Case 3
    cout <<a * b;
      break;
Case 4:
    cout a\b;
    break;
default:
    cout << " wrong input ";
 }
 return 0;
}
```

DOMS | Page No.
Date / /

Q 3) Odd or even

```
#include <iostream>
using namespace std;
int main()
{
int num, rem;
  cout << "enter number =";
     cin >> num;

  rem = num % 2;

    if (rem == 0)
  {
     cout << "num is even";
  }
  else
  {
    cout << "num is odd";
  }
  return 0;
}
```

for loop.

4) #include <iostream>.
using namespace std.
int main()
{
for (int i=1; i<=10; i++).
{
cout << i;
}
return 0;
}

While loop.

5) # include <iostream>.
using namespace std;

int main().
{
int i=1.
while (i<=10)
{
cout << i "\n";
i++;
}
return 0;
}.

31/7/23

DOMS | Page No.
Date / /

Experiment No. 1

Q Declare class students having data members as roll no, name, accept & display data for 1 object.

```
#include <iostream>
using namespace std
  class student
{
    int roll;
    String name;
public
    void accept()
{
cout << "Enter value of roll, name";
   cin >> roll >> name;
}
  void display()
  {
  cout << "roll:" << roll << "Name:" << name;
  }
};
   int main()
{
   Student S1;
   S1.accept();
   S1.display();
   return 0;
}
```

Q Write a program to declare a class book having data members as Book name, pages, accept for 2 objects & display the name of Book having greater price.

DOMS | Page No.
Date / /

2)
```
#include <iostream>
using namespace std;

class book
{
int pages;
String name;
  float price;

public:
void accept().
 {
cout << "Enter value of name, price, pages";
  cin >> name >> price >> pages;
}
 void display()
{
cout << "Name:" <<name<<", price" << price << pages
          <<pages:"
   }
     void display()
{.
cout  <<"name :" << name <<", prince :" << price
      << "Pages :" << Pages <<endl;
 }
};
 int main().
{
 book b1, b2;
  b1. accept();
  b2 : accept();
cout << "In Book with greater Price : \n";
```

```
if (b1.price > b2.price)
    {
      b1.  display();
    }
else if (b2.price>b1.price)
    {
      b2.  display();
    }
else
    {
cout << "Both books have the same
      price :\n";
      b1.display();
      b2.display();
    }
return 0;
}
```

DOMS | Page No.
Data / /

3) Write a program to declare a class time Accept the time in HH:MM:SS & convert it into seconds & display.

```
#include <iostream>
using namespace std;
class Time
{
 private
   int H, M, S;
public:
   void accept()
{
cout << "Enter Time";
   cin >> H >> M >> S;
}
 void display().
 {
  int total;
 total = ((H x 3600) + (M x 60) + S);
  cout << "Total time in seconds" << total;
  }
 };
int main()
 {
   time t1;
   t1. accept()
   t2. display();
  return 0;
}.
```

31/7/25

DOMS | Page No.
Date / /

Experiment NO. 2.

1) Write

```
#include <iostream>
  using namespace std;
  class city
{
 int population;
 string name;

public
 void accept()
{
cout<<" Enter city name & population";
  cin>> name>> population;
}
  void display().
{.
cout<<"name :"<<name<<": population:"<<population
      <<endl;
}
  int getpopulation()
{
  return population;
}
};
 int main()
 {
  city c[5];
for(int i=0; i<5; i++)
  {
```

```
    c[i].accept();
}
int max population=0;
for(i=1; i<5; i++)
{
if (c[i].getpopulation()>c[maxpopulation].getpopulation()
        max population=i;
}
}
cout<<"\ncity with highest population:\n";
    c[max population].display();
return 0;
}
```

3)
```
#include <iostream>
using namespace std;
class staff
{
String name;
String post;

public
void accept().
{
cout <<"Enter name and Post";
cin >> name >> post;
}
void display().
{
if (post == "HOD" || Post == "hod")
{
Cout << "HOD " << name << endl;
}
}
};
int main()
{
staff s[5]
for(int i=0 ; i<5; i++).
{
cout "\n enter details for staff "<i+1<<"\n";
```

```
    S[i].accept();
}
cout<<"\n Lists of HODs\n";
    for(i=0;i<5;i++)
        {
        S[i].display();
    }
    return 0;
}
```

31/7/25

DOMS | Page No.
Date / /

Experiment - 3

a Write a program to declare a class 'book' containg data members as book tiltle, aurthor name and price. Accept & display info of 1 object using this pointer.

```
# include <iostream>
  using namespace std;

class book
 {
  String b_title;
  String a_name;
   float price;

public:
 void accept().
{
 cout<<"enter book title" << endl;
 cin >> b_title;
 cout<<" enter aurthor name "<<endl;
  cin >> a_name;
 cout<<"enter price of book";
    cin >> price;
}
```

```
public:
  void display()
  {
  cout << " book nam: " << b_title <<endl;
  cout << " aurthor name: << a_name <<endl;
  cout << " price : " << price<<endl;
   }
};

int  main()
{
  book bl;
 book *p = & bl;
  p -> accept ();
  p-> display();

return 0;

}.
```

DOMS | Page No.
Date / /

WAP to declare class student having data member as roll no & percentage. Using this pointer invoke member function to accept the data & display this data for one object of a class.

```
#include <iostream>
using namespace std;
class student
{
  int roll;
  float per;
public:
void accept()
{
cout << "enter Student roll & percentage";
  cin >> this -> roll >> this -> per;
}
  void display()
{
  this -> accept();
cout << "roll = " << roll;
  cout << " percentage = " << per;
}
};
```

```
int main()
{
  Student S1;
    S1.display()
return 0;
}.
```

Output

enter student roll & percentage: 98 45

roll = 98 percentage = 45.

3) Write a program to demonstrate the use of nested class.

```
#include <iostream>
 using namespace Std;
  class Student {
        int roll;
      string name;
public:
    void accept()
  {
  cout << "enter roll & name;
      cin->> roll >> name;
  }
    void display()
  {
   Cout << "Name & rollno is" << name<<roll
        << endl;
  }
```

```
Class Marks {
   int m1, m2, avg
public
   Void accept()
   Cout << "enter marks m1 & m2";
        Cin >> m1>>m2.
    }.
  void Calculate_avg()
{
    avg (m1+m2)/2;
}
   void display(){.
   Cout<< "Average of marks is" <<avg<<endl
}
};
};
   int main()
  {
Student S1;
 S1. accept();
 S1. display();
Student:: marks sm;
sm. accept();
sm. calculate_avg();
 sm. display();
 Return 0;
};
```

14/8

Experiment - 4.

Q) WAP to swap 2 no's from same class using object as function argument.

```
#include <iostream>
Using namespace std;
Class Number {
  int num;
Public:
Void accept()
 {
 cout << "enter number";
    cin >> num;
 }
 Void display();
 {
 cout << "Number " << num << endl;
};
 Void Swap ( Number & obj)
 {
   int temp = num;
       num = obj.num;
       obj.num = temp;
 }
};
 int main()
{
Number n1, n2;
Cout << "enter first number " << endl;
      n1.accept();
  Cout << "enter second number" << endl;
```

```
    n2.accept();
    n1.swap(n2);

cout <<"\n. A fter swap:" <<endl;
    cout <<"first ";
        n1.display();

    cout <<" second ";
        n2.display();
    return 0;
}
```

Output

Enter first number : 6

Enter Second number : 45.

After Swap.

first number : 45

second number : 6.

WAP to swap 2 numbers from class using concept of friend function.

```
#include <iostream>
Using namespace std;
class temp {
    int n, y, q;
Public:
Void accept()
{
cout << "enter 2 numbers ";
    cin >> n >> y;
}
  void display ()
{
cout << "After swap x is ": <<n;
cout << "After swap y is ": << y ;
  }.
friend    void swap (temp &t);
};
  void swap (temp &t)
{
   t.q = t.x;
    t.x = t.y
    t.y = t.q
}
   int main()
{
   temp t1;
t1   .accept();
  swap (t1);
```

```
    ll.display();
    return 0;
}
```

Output

Enter two numbers: 9
8

After swap x is 6
y is 9.

DOMS | Page No.
Date / /

. WAP to swap two numbers from different class using concept of friend function

```
#include <iostream>
using namespace std;
class B;
class A {
int numA;
public:
void accept () {
cout << " enter number A : ";
cin >> numA;
}
void display () {
cout << " number A = " << numA << endl;
}
friend void swap numbers (A &, B &);
};
class B {
int numB;
public:
void accept() {
cout << " enter number B : ";
cin >> numB;
}
void display () {
cout << " number A : " << numB << endl;
}
friend void swap number (A &, B &);
};
```

DOMS | Page No.
Date / /

WAP to find the greatest number among 2 numbers from two different classes using friend function.

```
#include <iostream>
using namespace std;
class A {
  int a;
 Public :
 void accept()
{
cout << "enter a value:";
     cin>> a;
}
  Friend void greater (A a1, B b1);
};
  Class B;
 Public :
 Void accept(){
 cout <<"enter a value";
    cin >> b;
}
 Friend void greater (A a1, B b1);
};
void greater (A a1, B b1) {
      if (a1.a > b1.b) {
    cout <<" First value is greater";
    }
      else {
    cout <<" second value is greater";
```

```
};
};
int main()
{
  A x;
  B y;
      x.accept();
      y.accept();
        greater (x,y)
  };
```

DOMS | Page No.
Date / /

Experiment - 5

Q1 W.a.p to find the sum of numbers between 1 to n using a construction where the value

```
#include<iostream>
using namespace std;

class number {
    int num;
public
    number (){
    (out<<"Enter a number: "<<end l;
    cin>>num;

      int sum 0;
      for (int=1; i<=num; i++){
        sum = sum+i;
    }

    (out<<" sum of number upto <<num <<";
        : "<<sum <<end l;
    }
};

int main (){
    number;

    return 0;
}
```

→ output:
Enter a number: 3
The sum of numbers upto 3 is 6.

Q2 WAP to declare a class "student" having data members as name and percentage. Write a constructor to initialize these data members. Accept and display data for one student.

```
#include <iostream>
using namespace std;

class student {
    string name;
    float per;
    public:
    student(string n, float p) {
        name = n;
        per = p;
    }

    void display () {
        cout << "Name: " << name << endl;
        cout << "Percentage: " << per << endl;
    }
};

int main() {
    student s("Latasha", 93.2)
    s.display();

    return 0;
}
```

→ Output
Name: Letasha
Percentage: 93.2

Q3 Define a class "college" members variables as roll-no, name, course, WAP using constructor with default value as "computer Engineering" for course, Accept this data for 2 objects of class and display the data

```
#include <iostream>
using namespace std;

class college{
    int roll;
    string name;
    string course;
    public:
    College (int r=00, string n="unknown", string
        c="Computer Engineering"){
        roll=r;
        name=n;
        course=c;
    }

void display (){
    Cout<<"Roll number:"<<roll<<endl;
    cout<<"Name:"<<name<<endl;
    cout<<"course:"<<Cours<<endl;
};
```

```
int main (){
    college s1 (3) "Latasha");
    college s2 (18, "Gaurav");

    s1. display();
    s2. display();

    return 0;
}
```

→ Output

```
Roll numbers 37
Name : Latasha
Course : Computer Engineering
Roll number : 18
Name : Gaurav
course : Computer Engineering
```

Q4 WAP to demonstrate constructor over loading

```
# include <io stream>
using namespace std;

class student {
    int roll;
    string name;
    public:
    student (){
        name = "Unknown";
        roll = 0;
    }
```

```
Student () {
name = "Unknown";
roll = 0;
}

student (string n) {
name = n;
roll = 0;
}

student (string n, int r) {
name = n;
roll = r;

void display () {
cout "Name: " << name << endl;
Counter "Roll number: " << roll << endl;
}
};

int main () {
student s1;
student s2 ("Latasha");
student s3 ("Gaurav", 18);

s1. display ();
s2. display ();
s3. display ();
return 0;
```

→ Output:
Name : Unknown
Roll number : 0
Name : Latasha
Roll number : 0
Name : Gaurav
Roll number : 18

12/11

Experiment - 6

Q1 Write a program to implement multilevel inheritance. Assume suitable data.

```
#include <iostream>
using namespace std;

class department {
protected:
    string dname;
};

class student : protected department {
protected
    string sname;
    int roll;
};

class marks : protected student {
    int m1, m2, percentage;
public:
    void accept (){
        cout << "Enter department: " << endl;
        cin >> dname;
        cout << "Enter name: " << endl;
        cin >> s.name;
        cout << "Enter marks 1; " << endl;
        cin >> m1;
        cout << Enter marks 2: " << endl;
        cin >> m2;
```

```
void calculate(){
    int per = (m1 + m2) /2
    cout << "Department: " << dname)> ;
    cout << "Name: " << sname << end;
    cout << "Percentage: "<< per >> "%";
    }
};

int main () {
    marks m;
    m.accept ();
    m.calculate ();

    return 0;
}
```

→ output:

```
Enter department
Latasha
Enter marks 1;
88
Enter marks 2;
91
Department : CSE
Name: Latasha
Percentage :89
```

DOMS | Page No.
Date / /

Q2 WAP to implement multiple inheritance
Assume suitable data.

```
#include <iostream>
using namespace std;

class department{
protected:
    string dname;
};

class student{
protected:
    string sname;
    int roll;
};

class marks: protected department, protected
                                    student{
    int m1, m2, percentage;
public:
    void accept(){
    cout<<"Enter department:"<<endl;
    cin>>dname;
    cout<<"Enter name:"<<endl;
    cin>>sname;
    cout<<"Enter marks 1:"<<endl;
    cin>>m1;
    cout<<"Enter marks 2:"<<endl;
    cin>>m2;
```

```
void calculate(){
    int per=(m1+m2)/2;
    cout<<"Department: "<<dname>>
    cout<<"Name: "<<sname"nd1;
    (cout<<"Percentage: "<<percentage>>
    }
};

int main(){
    marks m;
    m.accept();
    m.calculate();
}
```

Q3 WAP to implement hierarchical inheritance

```
#include<iostream>
using name space std;

Class Person{
    protected:
    string name;
    intage;
    public:
    void acceptPer(){
    cout<<"Entername:"<<end!;
    (in >> name;
    (out<<"Enter age:"<<end1;)
    (in >> age;
}
```

```
class student : public Person {
    int roll;
    float per;
    public:
    void accept stud () {
        Cout << "Enter roll number " << end l;
        Cin >> roll;
        Cout << " Enter Percentage: " << end l;
        Cin >> per;
    }

    void display stud () {
        Cout << "Name : " << name << end l;
        Cout << " Age : " << age << end l;
        Cout << "Roll no : " << roll << end l;
        Cout << " Percentage : " << per << end l;
    }
};

class staff : public person {
        int emp. id;
        string subject;
        public:
        void accept staff () {
        Cout << " Enter employee ID: " << end l;
        Cin >> emp. id;
        Cout << " Enter subject: " << end l;
        Cin >> subject;
    }

    void display staff () {
```

```
        cout << "Name : " << name << endl;
        cout << "Age : " << age << endl;
        cout << "Emp ID : " << emp_id << endl;
        cout << "Subject taught : " << subject << endl;
    }
};

int main() {
    student s;
    staff t;

    s.accept_Per();
    s.accept_student();
    t.accept_Per();
    t.accept_staff();

    s.display_stud();
    t.display_staff();

    return 0;
}
```

Q4 Write a program to implement hybrid inheritance

```
#include <iostream>
using namespace std;

class college {
    protected:
        string cname;
};
```

```
class Employee : protected college {
    protected :
    string emp_name;
    int id;
};

class staff : public Employee {
    string name;
    int dept Id;
    public;
    void accept Emp (){
        cout << "Enter clg. name: " << endl;
        cin >> cname;
        cout << "Enter Emp. name: " << endl;
        cin >> emp_name;
        cout << "Enter I I) ; " << endl;
        cin >> id;
        Cout << Enter staff name: " << endl;
        Cin >> name;
        cout << "Enter dept id: " << endl;
        Cin >> dept Id;
    }

    Void display Emp (){
    Cout << "college: " << cname << end;
    Cout << "Emp name: " << emp. name;
    Cout << "ID: " << id << end;
    Cout << "staff name : " name;
    Cout "Department ID: " << dept Id >"
    }
};
```

```
class student : protected college {
        string stu_name;
        int roll;
        public:
        void accept stud (){
          cout<<"Enter college name : "<<endl;
          cin >> cname;
          cout<<" Enter name : "<<endl;
          cin >> stu_name;
          cout <<" Enter roll number : " <<endl;
          cin >> roll;
  }

void display stud () {
    cout <<" college : "<< cname << endl;
    cout <<" student name : "<< stu_name <<endl;
    cout <<" Roll number : "roll << endl;
}

int main (){
    Staff staff1;
    Staff 1. accept Emp ();
    Staff 1. display Emp ();

    student stud1;
    stud 1. accept stud ();
    Stud 1. display stud ();

    return 0;

}
```

e) WAP to demonstrate virtual base class Assume suitable data with figures.

```
# include <iostream>
using name space std;

class college students {
protected:
    int student_id;
    string ccode;
public:
    void accept ()
    cout << "Enter student ID: ";
    (in >> student_id;
    (out << "Enter college code: ";
    (in >> ccode;
    }

    void display (){
        (out << student ID: " << student_id << end
        (out << "college code: " << ccode << end)
    }
};

class test : virtual public college student {
protected:
    float percentage;
public:
    void accept (){
```

```
college student :: accept ();
cout << "Enter test percentage : ";
cin >> percentage;
}

void display () {
    college Student :: display ();
    cout << "Test percentage: " << percentage
                << "%" << endl;
}
};

class sports : virtual public college student {
protected:
    char grade;
public:
    void accept () {
    cout << "Enter sports Grade: ";
    cin >> grade;
    }

    void display () {
        cout "sports Grade: " << grade << endl;
    }
};

class Result: public test, public sports {
    float tot_marks;
public:
    void accept () {
        college student :: accept ();
```

```
cout <<"Enter test percentage : ";
cin >> percentage;
cout <<"Enter sports Grade : ";
cin >> ~~percentage~~ grade
cout <<"Enter ~~sports~~ total marks: ";
cin >>total_marks;
}

void display (){
    college student :: display ();
    cout "Test percentage " << percentage
    ~~cout <<~~ "~~sp~~ << "% " << endl;
    cout <<" sports Grade : " <<grade <<
    cout <<" Total marks : " <<tot_mak <<endl;
    }
};

int main (){
    Result r;
    cout <<"--- Enter student Details --- " <<endl;
    r.accept ();
    cout <<" \n ... student Result --- " <<endl;
    r.display ();
    return 0;
}
```

12/11

Experiment - 7

Q1 WAP using function over loading to calculate the area of a laboratory (rectangles) and area of class room (square)

```
#include <iostream>
using namespace std;

class Area {
public:
    float calculate (float length, float breadth){
      return length * breadth;
    }

    float calculate (float side){
      return side * side;
    }
};

int main (){
    Area a;
    Cout << "Area of laboratory : "<<a.calculate (10,5)
                                  <<endl;
    Cout "Area of square : "area.calculate (5)
                                  <<endl;
```

Q2 WAP using function over loading to calculate sum of 5 float values and sum of 10 integers values.

```
#include <iostream>
using namespace std;

class sum{
public:
    int total(int a[], int n){
        int s=0;
        for(int i=0; i<n; i++)
            s+=a[i];
        return s;
    }

    float total(float a[], int n){
        float s=0;
        for(int i=0; i<n; i++)
            s+=a[i];
        return s;
    }
};

int main(){
    sum s;
    int marks[10]={45,56,67,78,89,90,76,88,92,
                   85};
    float grades[5]={9.2,8.7,9.5,8.9,9.0};
```

```
cout << "Sum of 10 student marks: " <<
        S.total(marks, 10) << endl;
cout << " sum of 5, student grade parts : "
     << S.total(grades, 5) << endl;

return 0;
}
```

Q3 WAP to demonstrate the compile time and implement Unary operator when used with the object so that the numeric data member of the class is negated

```
#include <iostream>
using namespace std;

class Teacher {
    int experience;
public:
    Teacher(int e){
    experience = e;
    }
    void display(){
    cout << "Experience: " << experience << "
                    " year" << endl;
    }
    void operator-(){
    experience = -experience;
    }
};
```

```
int main (){
  Teacher t1(10);
  t1.display ();
  -t1;
  Cout << "After negation: ";
  t1.display ();
  return 0;
}
```

Q4 WAP to implement the Unary ++ operation when used with the object so that the numeric data member of the class is Incremented

```
#include <iostream>.
using name space std;

class student{
    int count;
public:
   student (int c=0){
      count = c;
   }
   void operation ++ (){
      ++count;
   }
   void operation ++ (int){
     count ++;
   }

   void display {
      cout << "student count: " << count << endl;
   }
```

```
};

int main () {
    student s1 (50);
    cout << "Before increment: " << endl;
    s1. display ();

    ++s1;
    cout << " After pre-increment: " << endl;
    s1. display ();

    s1++;
    cout << " After post-increment: " end;
    s1. display ();
    return 0;
}
```

12/11

DOMS | Page No.
Date / /

Experiment - 8

Q1 WAP to overload the '+' operator so that 2 strings can be concatenated.

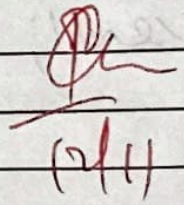
```
#include<iostream>
#include<string>
using namespace std;

class combine{
    string str;
public:
    combine(string s="")
        str=s;
    }

    Combine operator+(Combine &obj){
    Return combine(str+obj.str);
    }
    void display(){
        cout<<str<<endl;
    }
};

int main(){
    combine s1("xyz"), s2("pqr"), s3;
    s3=s1+s2;
    cout<<"concatenated String:";
    s3.display();
}
```

Q2 Write a program to create a base class I Login having data members name and pass-word. Declare accept() function virtual. Derive Email Login and membership login class from I Login. Display Email Login details and membership login details of the employee.

```
#include<iostream>
#include<string>
using namespace std;

class I login{
protected:
string name, password;
public:
    virtual void accept(){
    cout<<"Enter name:";
    cin>>name;
    cout<<"Enter password;";
    cin>>password;
  }
  virtual void display(){
    cout<<"Name:"<<Name<<name<<endl;
    cout<<"Password:"<<password<<endl;
  }
};

class Email Login : public I login{
    string email;
public:
    void accept() override{
```

```
        cout << "Enter Email ID: ";
        cin >> email;
        I Login :: accept ();
    }

    void display () override {
        cout "In ---- Email Login details --- "<<endl;
        cout <<"Email ID: " <<email << endl;
        I login :: display ();
    }
};

class membership login :public I Login {
        string member ID;
public:
    void accept () override {
        cout << "Enter Membership ID: ";
        cin >> member ID;
        I Login:: accept ();
    }

    void display () override {
        cout << "In --- membership login Details ---
                                    "<<endl;
        cout<<"Membership ID "<< member ID << endl;
        I Login :: display ();
    }
};

int main (){
    I Login * Login;
```

```
Email Login";
Membership Login *;

Login = &e;
Login -> accept ();
Login -> display ();

Login = &m;
Login -> accept ();
Login -> display ();

return 0;
}
```



Experiment - 9

Q1 Write a program to copy the content in one file into another. Open "First.txt"; read mode & "Second.text". Assume First text" is already created.

```
#include <iostream>
#include <fstream>
using namespace std;

int main(){
    ifstream infile("First.txt");
    ofstream outfile("second.txt");

    if(!infile){
     cout<"Error opening First.txt"<<endl;
     return 1;
    }

    char ch;
    while(infile.get(ch)){
      outfile.put(ch);
    }

    cout<<"File copied successfully"<<endl;

    infile.close();
    outfile.close();
    return 0;
}
```

Q2 WAP to Count Digits and Spaces using File Handling

```
#include <iostream>
#include <fstream>
using namespace std;

int main(){
    ifstream file("First.txt");
    if (file){
    cout << "Error opening File" << endl;
    return 1;
    }

    char ch;
    int digits = 0; spaces = 0;

    while (file.get(ch)){
        if (isdigit(ch))
            digit++
        else if (isspace(ch))
            spaces++;
    }

    (cout << "Digits:" << digits << endl;
    (cout << "Spaces:" << spaces << endl;

    file.close();
    return 0;
}
```

Q3 WAP to count words using file Handling;

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main () {
  ifstream file("First.txt");
  if (file) {
   cout << "Error" << endl;
   return 1;
  }

  string word;
  int count = 0;

  while (file >> word)
   count++;

  cout "Total words: " << count << endl;
  file.close()
  return 0;
```

Q4 Write a C++ program to count occurence of a given word using file handling

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
```

```
int main (){
  if stream file ("First.txt");
  if (!file)
  {cout<<"Error"<<endl;
  return 1;
  }

string word, target;
int count = 0;

cout <<" Enter word to count: "endl;
cin >> target;

while (file >> word) {
  if (word = target)
    count ++;
}

cout<<" Occurence of "<< target <<" is ".
          <<count <<endl;
  file.close ();
}
```

Q1

[2|1]

DOMS | Page No.
Date / /

Experiment 10

a) write a program to find sum and array elem using function template (eg pass integer, float and double array of 10 elements)

```
#include <iostream>
using namespace std;
template <class T> T func(T a[], int n){
    T sum=0;
    int i;
    for(i=0;i<n;i++){
        sum=sum+a[i];
}
return sum;
}
int main(){
  int a1[3]={2,0,2};
  float a2[3]={1,2,2,2,2,4};
  double a3[3]={50.5,20.5,30.5};
  cout<<"sum of integer array is:"<<sum(a1,
  <<endl;
  cout<<"sum of float array is:"<<func(a2,3)
  <<endl;
  cout<<"sum of double array is:"<<func(a3,3
  <<endl;
  return 0;
}
```

* Output

sum of integer array is: 4

sum of float array is: 4.8

Sum of double array = 6.1.5

b) Write a C++ program of square function using template specialisation :-
:- calculate the square of integer no & a string
:- Write a specialized function for the square of a string

```
#include <iostream>
using namespace std;
template <class T> T square (T x){
    T result
    result = x * x;
    return result;
}
template <> string square <string>(string ss ){
  return (ss + ss);
}
int main ( ){
    int i = 2;;
    string ww ("Aarya");
    ii = square <int> (i);
    cout << i << " :: " << ii << endl;
    cout << square <string> (ww) << endl;
    return 0;
}
```

* output
2 : 4
AaryaAarya

DOMS | Page No.
Date / /

Q) Write a c++ program to build simple calculator using class template

```
#include <math>
#include <iostream>
using namespace std;
template <class T>
class calculation {
private
  T num1, num2;
public:
  Calculator (Tn1, Tn2) {
    num1 = n1;
    num2 = n2;
}
T add () {
  return num1 + num2;
}
T sub () {
  return num1 - num2;
}
T mul () {
  return num1 * num2;
}
T divide () {
  if (num2 == 0) {
  (out << "error)" << endl;
  return 0;
  } return num1/num2;
}
void display numbers () {
(out << "numbers :" << num1 << num2 << endl;
};
```

```
int main (){
double a, b;
int choice;
Cout <<"enter 2 numbers: ";
Cin >> a >> b;
calculator <double> calc (a,b);
Cout <<" choose an operation ";
Cout <<" 1. Addition \n";
Cout <<" 2. substration \n";
Cout <<" 3. Multiplication \n";
Cout <<" 4. Division \n";
Cin >> choice
switch (choice){
Case 1:
  Cout << "Result :" << cal.add (a,b)<<endl;
  break;
Case 2
  Cout <<"Result",<< calc.sub(a,b)<<endl;
  break;
Case 3;
Cout <<"Result:" << calc.mul(a,b.) <<endl;
break
Case 4:
  cCout << "Result:" <<calc.div (a,b)<<endl;
  break;
  default:
  Cout <<" invalid choice "<<endl;
}
return 0;
}
```

12/11

Output
enter two numbers : 2,4
Choose an operation : 1
1. Addition
2. Substraction
3. Multiplication
4. Division
Result : 6

choose an operation : 3
1. Addition
2. substration
3. Multiplication
4. Division
Result : 8

Write a c++ program to implement push & pop methods from stack using class template

```
#include <iostream>
#include <vector>
using namespace std;
template <typename T>
class stack {
private:
    vector<T> elements;
public:
    void push(T item){
    elements.push_back(item);
    cout<<item<<"pushed to stack\n";
}
    void pop(){
```

```
    if (elements.empty()){
        cout << "Stack is empty. cannot pop in ";
        return;
    }
    cout << "elements.back()<<" popped in";
    elements.popback();
}
void display(){
    cout << "stack elements :";
    for(auto item : elements)
        cout << item << " ";
    cout << endl;
};

int main(){
    stack<int> s;
    s.push(10);
    s.push(20);
    s.push(30);
    s.display();
    s.pop();
    s.display();
    return 0;
}
```

* output

10 pushed to stack
20 pushed to stack
30 pushed to stack
stack elements : 10 20 30
30 popped
stack elements 10 20

DOMS | Page No.
Date / /

Experiment 11

a. Write a C++ program to implement generic ve
a) To modify the value of a given element.

```
#include <iostream>
using namespace std;
template <typename T>
class vector {
    T a[100];
    int size;
public:
    vector(int s): size(s){}
    void set(int i, T val){
    if(i>=0, && i<size)
    return a[i];
    cout<<"invalid";
    return T();
}
    void display(){
    for(int i=0; i<size; i++)
        cout<<a[i]<<" ";
        cout<<endl;
    }
};

int main(){
    vector<int> v(5);
    for(int i=0; i<5; i++)
        v.set(i, i*10);
    v.display();
    v.set(2, 99);
    cout<<"after modification:";
```

```
    v.display();
    return 0;
}
```

* Output
0 10 20 30 40
after modification: 0 10 99 30 40

b) To multiply by a scalar value

```
#include <iostream>
#include <vector>
using namespace std;
int main(){
    vector<int> vec = {1,2,3,4,5};
    int scalar = 3;
    for(int &val : vec){
        val = val * scalar;
    }
    for(int val : vec){
        cout << val << " ";
    }
    cout << endl;
    return 0;
}
```

* output
3 6 9 12 15

c) To display the vector in the form (10, 20, 30.....).

```
#include <iostream>
#include <vector>
```

```
using namespace std;
int main(){
    vector<int> vec = {10,20,30,40,50};
    cout<<vec[i];
    if(i!=vec.size()-1)
      cout<<",";
}
cout<<")"<<endl;
return 0;
}
```

* Output
(10, 20, 30, 40, 50)

12/11

Experiment : 12

Q Write a c++ program using STL.

a) Implement stack

```
#include <iostream>
#include <stack>
using namespace std;
int main(){
stack <int>s;
s.push(10);
s.push(20);
s.push(30);
cout<<"Top element :"<<s.top()<<endl;
s.pop();
cout<<"Top element after pop :"<<s.top()<<
endl;
cout<<stack size :"<<s.size()<<endl;
if(s.empty()){
  cout<<"stack is empty"<<endl;
}else{
  cout<<"stack is not empty";
}
return 0;
}
```

* Output

```
Top element : 30
Top element after pop : 20
Stack size : 2
stack is not empty
```

b) Implement queue

```
#include <iostream>
#include <queue>
using namespace std;
int main (){
    queue <int> q;
    q.push (10);
    q.pop (20);
    q.push (30);
    Cout <<"front elements :" <<q.front ()&(end
 Cout<<"back elements :"<<q.back ()<<endl;
q.pop ();

 Cout<<"after pop operation :"<<endl;
 Cout<<"front element :"<<q.front()<<endl;
 Cout<<"queue size :"<<q.size()<<endl;
 if (q.empty ()){
  Cout <<"queue is empty :"<<endl;
 } else {
    Cout<<"queue is not empty"<<endl;
 }
 return 0;
 }
```

* Output

front element : 10
back element : 30
after pop operation;
front element : 20
queue size : 2
queue is not empty

c) Implement sorting and searching with user-defined records such as personal record (name, birth date, telephone no), item record (item code, item name, quantity & cost).

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

struct Person {
    string name;
    int age;
    Person(String n, int a): name(n), age(a){}
};
int main(){
    vector<person> people = {
        Person("Alice", 30);
        Person("Bob", 25),
        Person("Charlie", 35);
        Person("David", 28) };

Sort(people.begin(), people.end(),
[](const Person& a, const person& b){
    return compare.Age(a,b); }

cout << "Sorted records by age: \n";
for(auto &p: people){

cout << p.name << "-" << p.age << "in";
}
```

```
int searchAge = 28;
int foundIndex = -1;
for (int i = 0; i < (int) people.size(); i++) {
    if (people[i].age == searchAge ? 1 : 0) {
        foundIndex = i;
        break;
    }
} if (foundIndex != -1)
    cout << "person with age " << searchAge <<
"found" << people[foundIndex].name << "\n";

else
    cout << " person with age " << searchAge <<
"not found";
return 0;
}
```

* Output

Sorted records by Age:

Bob - 25

David - 28

Alice - 30

Charlie - 35

Person with age 28 found: David.

12/11

DOMS | Page No.
Date / /