PAGE No.
DATE / /

1) Write a c++ programme for addition of 2 number

```
# include <iostream.h>

void main ()
{
int a,b,c;

cout << "enter no a & b";

cin >> a >> b;

cout << " addition = " << c;
}
```

output:-
enter 2 number's
10 12
addition = 22

2) Write a C++ code to perform arithmatic operation using switch case.

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

 int main ()
{
  Char op;
  Float num1 ,num2;
  cout << "enter operator : +, -, *, / :";

  cin >> num1 >> num2;

  switch (op) {
    case '+' :
cout << num1 << " +" << num2 << " =" <<
                                num1 + num2;
  break ;

  case '-' :
cout << num1 << " -" << num2 << " =" << num1
                                - num2;
  break;

  case '*' :
```

```
cout<<num1<<" * "<<num2<<" = "<<num1
                                * num2;
break;

case '/';
cout<<num1<<" / "<<num2<<" = "<<num1
                                / num2;

break;

default:
cout<<"error operator not found";
break;
}
  return 0;
}
```

output

enter operator : +, -, *, / : +

enter 2 number's 12 12

12 + 12 = 24

PAGE No.
DATE / /

3) Programme to check number is even or odd

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

int main ()
{
  int a ;

  cout << "enter a number";
  cin >> a;

  if ( a % 2 ==0)

  cout << a << "number is even";
  else
  cout << "number is odd";

  return 0;
}
```

out put -
enter a number 25
number is odd

PAGE No.
DATE / /

4) C++ code to print 1 to 10 number using for loop.

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

int main ()
{
    for (int i = 1 ; i<=10 ; i++)
    {
        cout << i << " ";
    }
    return 0;
}
```

PAGE No.
DATE / /

5) c++ code to print 10 to 1 number using while loop

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

int main ()
{
   int i =10;

   while (i >=1) {
      count << i << " ";
      i --;
}
   return 0;
}
```

output
10
9
8
7
6
5
4
3
2
1

number casing

6) C++ code to print below pattern

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

int main ()
{
   int i, j;

   for (i=0; i<6; i++)
   {
      for (j=1; j<i; j++)
      {
         cout <<j;
      }
      cout << "\n";
   }

   return 0;
}
```

Output-

```
1
1 2
1 2 3
1 2 3 4
```

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

int main ()
{
    int rows = 5;
    for (int i = 1; i<=rows; i++)
    {
        for (int j = 1; j<=i ; j++)
        {
            cout << j << " ";
        }
        cout << endl;
    }
    return 0;
}
```

output

```
1
2   2
3   3   3
4   4   4   4
5   5   5   5   5
```

PAGE No.
DATE / /

```
)
ps; i++)
=1 ; j++)
" ";
```

PAGE No.
DATE / /

c) #include <iostream>
using namespace std;

```
int main ()
{
    int rows = 5;
    int space ;

    for (int i = 1; i<=rows ; i++)
    {
        space = rows - i;
        for (int s=1 ; s<= space ; s++)
        {
            cout << " ";
        }
        for (int j =1 ; j<= i ; j++)
        {
            cout << "*";
        }
        cout << endl;
    }

    return 0;
}
```

8/7/25

PAGE No.
DATE / /

output -

```
        *
      *   *
    *   *   *
  *   *   *   *
*   *   *   *   *
```

PAGE No.
DATE / /

Experiment - 1

1) Write a C++ programme to declare a class student having data member's as roll_no & name. Accept and display data for single student.

code -

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

class student
{
 int roll_No;
 string name;
 public :

void accept ()
{
cout <<"enter student name & roll_No\n";
cin >> Name >> rollNo;
}

void display ()
{
cout <<" name of student :" <<name;
cout <<"\n roll No of student : " <<rollNo;
}
```

```
};

int main()
{
  student s1;
  s1.accept();
  s1.display();
  return 0;
}
```

output -

```
Enter student Name & roll_NO
bhushan 2 23
Name of student : bhushan
Roll No of student : 23
```

2) Write
having
5pages
display
price.

code:

```
# incl
using

class
{
  publ
int t
string
public
void
{
cout<<
cin>>
}
void
{
cout<<

}
};
int
```

2) Write a c++ code to crear a class book having data member's as bname, bprice, bpages. Accept the data for 2 book and display the name of the book having greater price.

code:

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

class classbook
{
 public:
 int bprice, bpages;
 string bname;
 public:
 void accept ()
 {
 cout<<"enter book name, price and pages;
 cin>>bname>>bprice>>bpages;
 }
 void disp()
 {
 cout<<"the name of the most expensive book
                    is"<<bname;

 }
};
int main ()
```

```
{
class book b1, b2;
b1.accept();
b2.accept();

if(b1.bprice > b2.bprice)
{
  b1.disp();
}
else
{
  b2.disp();
}
  return 0;
}
```

output -

Enter book name, price and pages : abc 1006
Enter book Name, price and pages : cdc 50 b
The name of the most expensive book is abc.

3) Write a pr
accept time
it to total

code:

```
# include
using nam
class time
{
 Public:
 int hours
void acce
{
cout << "e
cin >> hou
}

void di
{
cout << "
min << "
int sh =
int sm =
int tt = s
cout << "
  }
};
```

3) Write a programme to declare class "time", accept time in HH:MM:SS format, convert it to total second's and display them.

code.

```
# include <iostream>
using namespace std;
class time
{
 public:
 int hours, min, sec;
 void accept ()
 {
 cout << "enter time in hh mm ss";
 cin >> hours >> min >> sec;
 }

 void disp ()
 {
 cout << "the time in format: << hours << ":" <<
 min << ":" << sec << endl;
 int sh = hour * 3600;
 int sm = min * 60;
 int tt = sh + sm + sec;
 cout << " total times tt is << tt;
  }
 };
```

PAGE No.
DATE / /

```
int main ()
{
class time t;
t accept ();
t. disp ();
return 0;
}
```

output -
enter time in hh mm ss 5 35)
the time time in format is 5 : 35 :)
total time in seconds 20100)

15/7/25

PAGE No.
DATE / /

22.7.25

Experiment - 2

1) WAP to declare a class city having data members as name and population. Accept this data for 5 cities and display name of city having highest population.

code-

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

class city
{
  public :
  string name;
  int population;

void accept ()
{
cout << "enter the name of 2 population of
          5 cities";
cin >> name >> population;
}

void display ()
{
  cout << " the name of the most popula-
          red city is ";

}
```

PAGE No.
DATE / /

```
};

int main ()
{
    city c[5];
    int i;
    for(i=0; i<5; i++)
    {
        c[i].accept(c);
    }

for(i=0; i<5; i++)
{
    c[i].display();
}

for (int i=0; i<5; i++)
{
    int j;
    for(j=0; j<5; j++)
    {
    if (c[j].population>c[i].population)
    {
        break;
    }
}
if (j==5)
```

```
{
  cout<< "\n the most populated cit is"
    << c [i] name
    << "with population" << c [i] population
                                << endl;
  }
}
 return 0;
}
```

output -

Enter the name & population of 5 cities

Delhi 100

mumbai 200

Pune 300

bangloe 400

Nagpure 50

The most population city is baglore

2) WAP to declare a class account having data member as account no and balance. accept this data for 10 account and give interest of 10% where balance is equal or greater than 5000 and display them.

code.

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

class account
{
 public:
 int accountNo;
 float balance;

 void accept ()
 {
  cout<<"Enter account number";
  cin>>accountNo;
  cout<<"Enter balance";
  cin>>balance;
 }

 void display ()
 {
  cout<<"Account No:"<<accountNo<<",
     Balance:"<<balance<<endl;
 }
```

```
};
int main ()
{
  Account acc [10];
cout << "Enter detail for 10 accounts \n";
for( int i =0; i<10; i++)
{
 cout << "\n Account" << i+1 << ": \n";
 acc [i] . accept ();
}
for (int i=0; i<10; i++)
{
  if (acc [i]. balance >=5000)
  {
    acc [i]. balance += acc [i] . balance * 0.15;
  }
}
cout << "\n updated account Detail: \n";
for (int i=0; i<10; i++)
{
   acc [i]. display ();
}
return 0;
}
```

output -
Enter details for 10 account

1) Account number : 1
Balance : 6000

2) Account No : 2
Balance : 24000

3) Account No : 3
Balance : 65000

4) Account No : 4
Balance : 69000

5 Account No : 5
Balance : 5600

6 Account No : 6
Balance : 6500

7 Account No : 7
Balance : 5400

8 Account No : 8
Balance : 25000

9 Account No : 9
Balance : 4000

10) Account
Balance

Updat
1 66
2 26
3 71
4 75
5 61
6 71
7 50
8 2
9 49
10 11

10) Account No: 10
Balance: 10000

Updated account details

1	6600
2	26400
3	71500
4	75900
5	6100
6	7150
7	5940
8	2750
9	44000
10	11000

3) WAP to declare a class staff having data member's as name and post. accept this data for 5 staff and display names of staff who are HOD.

code.

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

class staff
{
  Public:
  string name;
  string post;

  void accept ( )
  {
   cout<<"Enter staff name :";
   cin >> name;
   cout << "Enter staff post :";
   cin >> post;
   }
  };

int main ( )
{
staff / staff Array [5];
```

having
post.
and display

```
cout<<"Enter details of 5 staff member:\n";
for(int i=0; i<5; i++)
{
 cout<<"\n staff "<<i+1<<"\n";
 staffArray[i].accept();
}

cout<<"\n staff member's who are HOD
                                :\n";
for(int i=0; i<5; i++)
{
 if(staffArray[i].Post =="hod")
 {
   cout<<staffArray[i].name<<endl;
 }
}
return 0;
}
```

output -

Name : abc
Post : HR

Name : abd
Post : CEO

Name : bcd
Post : Clerk

Name : XYZ
Post : Hod

Name : ZYX
Post : Hod

Staff member who are Hod : XYZ
ZYX

29/7/25

29.7.25

1) WAP
ning c
author
display
using

code
Find
using

class
public

void
{
co
cin
cou
cin
cou
cin
3

29.7.25

Experiment:-3

1) WAP to declare a class book containing data member's as book title, author name and price. Accept and display the information for one object using a pointer to that object

code-

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

class book {
public:
    string title;
    string author;
    float price;

void accept ()
{
 cout << "Enter book title :";
 cin >> title;
 cout << "Enter author name :";
 cin>>author;
 cout << "Enter price :";
 cin >> pripce;
}
```

XYZ
ZYX

```
void display ()
{
cout << "\n Book title : " << title;
cout << "\n Author Name : " << author;
cout << "\n Price : " << price;
}

};

int main ()
{
 Book * ptr = new Book;

 ptr -> accept ();
 ptr -> display ();

 delete ptr;
 return 0;
}
```

O/P -

```
Enter book title : abc
Enter ~~bee~~ author : bcd
Enter price : 500

Book title : abc
Author name : bcd
Price : 500
```

2) WAP to declare a class student having data member's as roll_no, percentage. using this pointer invoke member function to accept & display this data for one object of the class

code-

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

class student
{
  int roll_no;
  float percentage;
  Public:

  void accept ()
  {
    cout << "Enter your roll No";
    cin >> roll_No;
    cout << "Enter your percentage";
    cin >> percentage;
  }

  void display ()
  {
    cout << "Roll No: " << this -> roll_No << endl;
    cout << "Percentage : " << this -> percentage;
  }
};
```

```
int main ()
{
    student s1;
    s1.accept ();
    s1. display ();
    return 0;
}
```

O/P.

Enter your roll No: 23
Enter your percentage: 98
Roll No = 23
Percentage = 98

3) Demonstrate the use of nested loop

code-

```
#include <iostream>
using namespace std;
class student {
 public :
 class marks {
 int marks;
 public:
 void accept () {{
 cout << "write the mark for the class";
 cin >> marks;
 }
 void display () {
 cout << "the marks are" << marks;
 }
 }
};

int main () {
student :: marks m1
m1. accept ();
m1. display ();
return 0;
}
```

loop

class";

;

o/p.
Write the marks for class : 50
The marks are : 50

5/8/25

Experiment 4

1) Prog to swap 2 number from same class using object as function arguments. write swap function as member function

```
-> # include <iostream>
using namespace std;
class num {
public:
    int n1,n2
void accept ()
cout<< "enter first number:" <<endl;
cin >> n1;
cout << "enter second number:" <<endl;
cin >>n2;
}
void display () {
cout << "num 1:<<n1<<endl;
cout << " num 2: <<n2 << endl; }
friend void swap (num) };
void swap (num) {
int temp = n1; n1=n2 ; n2=temp;
cout<<"swapped :\n First number :<<n1
<<"\n second number <<n2
}
```

```
3 n4;
ing main
n1. accept
n2. disp
n1. swap
return
}

o/p -
Enter fi
Enter se

n1 : 5
n2 : 9

swapped
First n
second
```

PAGE No.
DATE / /

```
rom same
n arguments.
ember func-

" <<endl;

" <<endl;

; 3

temp;
ber :<<n1
```

PAGE No.
DATE / /

```
3 n1;
int main ()E
n1.accept ();
n2.display ();
n1.swap (n1);
return 0;
3
```

O/P -

```
Enter First number :5
Enter second number :9

n1 :5
n2 :9

swapped
First number :9
second number :5
```

2) swap 2 no from same class using concept of friend function

```
-> # include <iostream>
using namespace std;
class num {
public :
    int n1,n2;
    void accept ( ){
cout<< "enter first number : "<<endl;
cin >>n1;
cout<< " enter second number "<<endl;
cin >>n2;
}
void display ( ){
cout<<" num 1: << n1 << endl;
cout <<" num 2: <<n2 <<endl; }
friend void swap (num)};
}nu;
void swap (num){
int temp =nu.n1;
nu.n1 = nu.n2;
nu.n2 = temp;
cout<<" swapped : \n First : "<< nu.n1<<" \n
                    second : nu. <<endl
}
```

```
int main
nu. accept
nu. display
swap (n
return 0
}

op
enter
enter

num 1
num 2
swapped

First
second
```

PAGE No.
DATE / /

```
int main ()
{
n1.accept ();
n1.display ();
swap (n1);
return 0;
}
```

O/P

enter 1st number : 5
enter 2nd number : 8

num 1 : 5
num 2 : 8
~~swapped~~

First : 8
second : 5

3) swap 2 no from different class using friend function..

```
-> # include <iostream>
using namespace std;
class num 1 {
public :
        int n1;
void accept ()
cout<< "enter first number :"
cin >>n1;
}
void display ()
cout << " num1 : "<<n1<<endl
} num 1;
class num2 {
Public :
        int n2;
 void accept () {
cout << " enter second number : "<<endl
cin >> n2; }
void display ()
cout<< " num2 : "<<n2 <<endl;
} num2;
void swap (num1, num2)
int temp = num1.n1;
num1.n1 = num2.n2
num2.n2 = temp;
```

class cking

<<endl

```
cout << " swapped : \n first << num1.n1 << \n
        second : << num2.n2 << endl;
}
int main ()
{
num1.accept ();
num2. accept ();
num1. display ();
num2 . display ();
swap (num1, num2);
return 0;
}
```

o/p -

enter first number : 5

enter second number : 8

num 1 : 5

num 2 : 8

swapped

First : 8

second : 5

4) Create 2 class result and result 2 class stores the mark at the students read the value at mark both the class object is compute avg.

```
-> # include <iostream>
using name space std;
class result 1{
public:
    int v1;
  void accept ()
cout<<"enter first result:"<<endl;
cin>> v1; }
} res1;
class result 2{
public:
 int v2;
void accept ()
cout<<" enter second result "<<endl;
cin >>v2;
}res2;
float avg (result 1.result 2){
    return (res.v1 res.v2);
}
int main () {
 res1 . accept ();
 res 2 . accept ();
float avg =avg (res1, res2);
cout<<" Average : <<avg;
 return 0;
}
```

PAGE No.
DATE / /

O/P.
enter first result:
95
enter second result:
99
Average: 97

5) Find the greatest number among 2 numbers from 2 different classes using friend function

→ # include <iostream>
using namespace std;
class num1
public:
int n1;
void accept ()
cout << "enter 1 number " << endl;
cin >> n1; }
};
class num2 {
public
int n2;
void accept ()
cout << " enter 2 number " << endl;
cin >> n2; }
};
int main ()
n1 . accept ();
n2 . accept ();
int gr = gnum (n1 . n1 ; n2 . n2);
cout << " Greatest : " << n1 << endl;
return 0;
}

2 numbe
riend

PAGE No.
DATE / /

O/P
enter first number : 5
enter second number : 8
Greatest : 8

6) Create 2 classes class A and class B, with a private interpreter. Write a friend function sum () that can access private data from both the class and return the sum

```
-> #include <iostream>
using name space std;
class B;
class A {
int n1;
public:
void accept ()
{
cout<<enter number 1:";
cin >> n1;
}
friend int sum (A, B);
};
class B {
int n2;
public:
void accept ()
{
cout << " enter no. 2";
cin >>n2;
}
friend int sum(A, B)
};
```

```
int sum (B num1, B num2)
{
  return num1.n2 + num2.n2;
}
int main ()
{
  B obj 1;
  B obj 2;
  obj 1. accept ();
  obj 2. accept ();
  cout<< "sum is"<<sum (obj 1, obj 2)<<
                                  endl;

  return 0;
}
```

o/p:
enter number 1: 5
enter number 2: 4
sum is 9

7 WAP with class number that contains private integer use a friend function swap number to swap the private values of 2 number object.

```
# include <iostream>
using namespace std;
class number
{
int n;
public:
void accept ()
{
cout << "enter a number,";
cin >>n;
}
void display ()
{
cout << " \n num : " <<n;
}
friend void swap (number & num1, number
                  & num2);
};
void swap (number & num2, number & n
{
int temp = num1.n;
num1.n = num2.n;
num.2n = temp;
```

```
{
int main ()
{
number n1;
number n2;
num1. accept ();
num2. accept ();
cout << "original value :\n";
num1. disp ();
num2. disp ();
swap (num1, num2);
cout << "\n swapped values are :\n";
num2. disp ();
num1. disp ();
return 0;
}
```

O/P :

```
enter number : 5
enter number : 5
original values are
num 1 is 5
num 2 is 5
swapped value are
num 1 is 5
num 2 is 5
```

5) Define 2 class box and cube having private volume. Write a friend function kind greater box, cube that determines which object has a larger volume

```
# include <iostream>
using namespace std;
class cube;
class box
{
 int volume;
 public:
void accept()
{
cout << "enter volume of box";
cin >> volume;
}
friend void kindgreater (box vol1, cube v
};
class cube
{
 int volume;
 public:
void accept()
{
cout << "enter volume of cube";
cin >> volume;
}
```

```
friend void find greater (box vol 1, cube vol 2)
};
void find greater (box vol 1, cube vol 2)
{
 if (vol 1. volume > vol 2. volume)
{
cout << " enter volume of box is greater;
}
 else
{
cout << " volume of cube is greater;
 }
}
int main ()
{
box vol 1;
cube vol 2;
vol 2 .accept()
vol 1 . accept ()
find greater (vol 1, vol 2)
return 0;
}
```

O/P.

```
enter volume of box : 56
enter volume of cube : 22
volume of cube is greater
```

```
a) # include <iostream>
using name space std;
class complex
{
  int real;
  int imaginary;
  public:
  void accept()
  {
  cout<<"enter real part";
  cin >> real;
  cout<<enter imaginary part";
  cin >> imaginary;
  }
  friend void add (complex n1, complex n2
  };
  void add
  {
  int sum_real = n1.real + n2.real;
  int sum = imaginary = n1.imaginary + n2.imaginary;
  cout<<"the complex no<< sum.real<<"+"
  <<sum.imaginary<<"i";
  }
  int main ()
  {
  complex n1,n2;
  n1.accept ();
  n2.accept ();
  add (n1,n2);
  return 0;
```

PAGE No.
DATE / /

3

o/p -
enter real part : 6
enter imaginary : 5
enter real : 3
enter imaginary : 6
Final complex no : 9+14i

10) #include <iostream>
using namespace std;
class student
{
string name;
int m1, m2, m3;
public:
void accept ()
{
cout << "enter name of student";
cin >> name;
cout << "enter mark for 3 subject";
cin >> m1, >> m2 >> m3;
}
};
void average (s1)
{
int avg = s1.m1 + s1.m2 + s1.m3) / 3;
cout << "avg is: << avg;
}
int main ()
{
student s1;
s1. accept ();
s1. display ();
average ;
}

PAGE No.
DATE / /

o/p :
enter name of student : Bhushan
enter mark's for 3 subject : 70, 80, 45
average : 69

11 # include <iostream>
using namespace std;
class Beta;
class Gamma;
class alpha;
{
int n;
public:
void accept ()
{
cout << "enter value of alpha";
cin>> n;
}
friend void sum
};
class Beta
{
int n;
public:
void accept ()
{
cout << "enter value of Beta";
cin>> n;
}
friend void sum
};
class Gamma
{
int n;
public :

```
void accept()
{
cout << enter value of gamma";
cin>>n;
}
friend void sum
};
void sum
{
  int s=A.n+B.n+C.n;
cout<<" the sum is" << s;
}
int main()
{
 alpha A;
 Beta B;
 Gamma C;
 A.accept();
 B.accept();
 C.accept();
 sum (A,B,C);
 return 0;
}
```

```
op
enter value of alpha:5
enter value of Beta:6
enter value of gamma:7
the sum is :18
```

12) # include <iostream>
include <cmath>
using name space std;
class point {
int x, y;
public:
void accept()
{
cout << enter x value: ";
cin >> x;
cout << enter y value: ";
cin >> y;
}
friend void calculate (point p1, point p2);
};
double distance = sqrt (p2.x - p1.x) * p2.x +
+ (p2.y - p2.y) * (p2.y - p1.y)
cout << " the distance between the 2 points
is : " << distance;
}
int main()
{
point p1, p2;
p1.accept();
p2.accept();
calculate (p1, p2);
return 0;
}

o/p:
enter
enter
enter
enter
the d

PAGE No.
DATE / /

O/P:
enter x value : 3
enter y value : 6
enter x value : 3
enter y value : 7
the distance between them : 3.23607

13) # include <iostream>
using namespace std;
class audit;
class bank account
{
string name;
int bank balance;
public:
void accept ()
{
cout << enter";
cin >> name;
cout << " enter bank balance";
cin >> bank balance;
}
friend audit;
};
class audit
{
public:
void start Audit (bank account b1)
{
cout << name" << b1.name;
cout << bank balance" << bank balance;
}
};
int main ()
{

```
bank account b2;
audit A1;
b1.accept();
A1.start audit(b1);
return 0;
}
```

O/P

```
enter name : bhushan
enter balance : 10,000
name : bhushan
bank balance : 10,000
```

12/8

Experiment -5

1) WAP to find the sum of number's betw 1 to n using a constructor where the value of n will be passed to the constructor

Defaulte constructor-

```
#include <iostream>
using namespace std;
class sum
{
  int n;
  public:
  sum ()
  {
   cout<<"enter the value of n:";
   cin >> n;
  }
  void display ()
  {
   cout<<" sum from 1 to "<<n<< " is":
        << (n* (n+1))/2 <<endl;
  }
};
```

int main
{
sum
s1. di
return
}

o/p-
enter
sum r

```
int main ()
{
    sum s1;
    s1.display ();
    return 0;
}
```

o/p:

enter the value of n: 5

sum from 1 to 5 is : 15

Parameterised constructor

```
# include <iostream>
using namespace std;
class ~~std~~ sum
{
  int n;
  public:
  sum (int num)
  {
     n = num;
  }
  void display ()
  {
   cout<<" sum from 1 to "<<n<<" is "<<
                (n * (n+1)) /2<<endl;
  }
};
int main ()
{
  int num;
  cout<<" enter the value of n:";
  cin>> num;
  sum s (num);
  s. display ();
  return 0;
}
```

O/P -
enter the value of n: 5
sum from 1 to 5 is : 15

copy constructor

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

class sum
{
    int n;
    int sum;
    public:
    sum ()
    {
        n=4;
        sum=0;
    }
    sum (const sum &obj)
    {
        n= obj.n;
        sum=0;
        for( i=1 ; i<=n; i++)
        {
            sum += i;
        }
        cout << " sum= " << sum << endl;
    }
};
```

```
int main()
{
    sum s1;
    sum s2(s1);
    return 0;
}
```

O/P.
sum = 10

2) CPP to declare a class student having data member's as name and percentage write a constructor to initialize these data member. Accept and display data for one student.

Default constructor.

```
# include <io stream>
using namespace std;
class student
{
 string name;
 float percentage;
 public:
 student ()
 {
   name = "bhushan";
   percentage = 90;
 }
 void display ()
 {
  cout<<" Name :" << name;
  cout<<"\n percentage:" << percentage;
 }
};
int main ()
{
  student s1;
  s1.display ();
  return 0;
}
```

PAGE No.
DATE / /

orp -
Name : bhushan
percentage : 90

Parametrised-

```
#include <iostream>
using namespace std;
class students
{
  string name;
  float percentage;
  public:
  students (float p, string n)
  {
    percentage = p;
    name = n;
  }
  void display ()
  {
  cout << "percentage : " << percentage;
  cout << " \n Name : " << name;
  }
};
int main ()
{
  student s1 ( 90, "Bhushan");
  s1. display ();
  return 0;
}
```

O/P -

Percentage : 90

Name : Bhushan

copy.

```
# include <iostream>
# include <string>
using namespace std;
class student
{
    string name;
    float percentage;
    public:
    student ()
    {
      cout<<"enter name:";
      cin >> name;
      cout << " enter percentage :";
      cin>> percentage;
    }
    student (const student & obj)
    {
      name = obj.name;
      percentage = obj.percentag
    }
    void display ()
    {
      cout<<" Name:"<<name<<endl;
      cout<<" percentage :"<<percentage <<endl;
    }
};
```

```
int main ()
{
 student s1;
 student s2 (s1);
 cout << "\n student 1 details:\n";
 s1. display ();
 cout << "\n student 1 details copied:\n";
 s2. display ();
 return 0;
}
```

O/p -

student 1 details

name : bhushan

percentage : 99

student 1 details (copied)

name : bhushan

percentage : 99

3) Define a class college having data members as name, roll NO, course: OOP using constructor with default value as computer engineering for course. Accept this data for 2 objects for class and display the data.

```
# include <iostream>
using namespace std;
class college
{
    int roll_no;
    string name, course;
    public:
    college (int roll, string name, string course
                                    = computer
                                    engineering)

    {
        int r=roll;
        string n=name;
        string c=course;
        cout<<r<<n<<c;
    }
};
int main ()
{
    college c1 (1, "Krishnan");
    return 0;
}
```

PAGE No.
DATE / /

o/p.
2 bhushan computer engineering.

4) OOP to demonstrate constructor overloading.

```
# include <iostream>
# using namespace std;
class students
{
    string name;
    int roll No;
    float percentage;
    Public:
    students ()
    {
     name = "bhushan";
     roll No = 23;
     percentage = 89;
    }
    students (string n, int r)
    {
        name = n;
        roll No = r;
        percentage = 89;
    }
    students (string n, int r, float p)
    {
        name = n;
        roll No = r;
        percentage = p;
    }
```

```
void display ()
{
    cout<<" Name : "<<name
    cout<<" \n Roll No : "<<rollno
    cout<<" \n percentage : "<<percentage<<endl
    }
};
int main ()
{
    student s1;
    student s2("arjun", 26);
    student s3 ("shreyas, 21, 88.5);
    s1.display ();
    s2.display ();
    s3.display ();
    return 0;
}
```

O/P.

Name : bhushan
Roll No : 23
Percentage : 99

Name : arjun
Roll No : 26
percentage : 89

Name : shreyas
Roll No : 21
percentage : 95

16/9/25

Experiment 6

1) Single inheritance

Create a base class called person with attributes name and age. Derive a class students from person that add's an attribute roll No. Write function to display all detail's of the student.

Code-

```
#include <iostream>
using namespace std;
class person
{
    protected:
    int age;
    string name;
};
class students : public person
{
    int roll_No;
    public:

 void accept ()
{
  cout << "enter age:";
  cin >> age;
  cout << "enter name:";
  cin >> name;
```

```
enter cout<<" enter roll No : ";
cin>> roll_No;
}
void display ()
{
  cout<<" your age = "<<age;
  cout<<" \n your name = "<< name;
  cout<<" \n your roll No =<< roll_No;
}
};
int main ()
{
  student s1;
  s1 . accept ();
  s1 . display ();
  return 0;
}
```

o/p

```
enter age : 12
enter name: bhushan
enter roll No : 23
your age = 12
your name = bhushan
your roll No = 23
```

2) Multiple inheritance

create 2 base classes academic's and sport's
Academic's class contain marks of a stu-
dent
sport class contain sport score create a
derived class result inherit's from both
academic's and sport's write function to
calculate the total score and display details.

code:

```
# include <iostream>
using namespace std;
class academic's
{
    protected
    int marks;
};
class sport
{
    protected
    int score;
};
class result : protected academic's, protected sport
{
    int total_score;
    public :
```

```
void accept ()
{
  cout<<" enter mark : ";
  cin>> mark;
  cout<<" enter score : ";
  cin>> score;
}
void calculate ()
{
  total_score = mark + score
  cout<<" Total score = "<< total_score;
  }
};
int main()
{
  result r1;
  r1.accept ();
  r1.calculate ();
  return 0;
}
```

O/P

enter mark : 80

enter score : 20

3) Multilevel

create a class vehicle with attributes like brand and model Devire a class car from vehicle which add an attribute type further devire a class electric car from car which add battery capacity write function to display all the delail's.

code-

```
#include <iostream>
using namespace std;
class vehicle
{
  protected :
  string brand;
  string model;
};
class car : protected vehicle
{
    protected:
    string type;
};
class electric : protected car
{
    int battery_capacity;
    public:
    void accept ()
  {
```

```
cout<<"enter brand : ";
cin>>brand;
cout<<"enter model";
cin>>model;
cout<<"enter type : ";
cin>>type;
cout<<"enter battery cap:";
cin>> batery_capacity;
}
void display ()
{
cout<<"\n Brand ="<<brand;
cout<<"\n Model ="<<model;
cout<<"\n Type ="<<type;
cout<<"\n Battery capacity ="<<battery_capacity
}
};
int main ()
{
electric e1;
e1.accept ();
e1.display ();
return 0;
}
```

op-

```
enter brand : ABC
enter model : BCD
enter type : SUV
enter battery cap : 250cc
```

Brand = ABC
Model = BCD
Type = SUV
Battery capacity = 25 000

30/9/25

PAGE No.
DATE / /

4) Hierarchical

create a base class employee with attributes empid and name. Derive 2 classes manager and developer from employee. Manager has an attribute department and developer has an attribute programming language. write function to display detail for both.

code-

```
#include <iostream>
using namespace std;
class employee
{
  protected
  int emp ID = 23;
  string name = "bhushan";
};
class manager : protected employee
{
   string dept = "poly";
   public:
   void display ()
   {
     cout<<"\n emp id is :"<< empID;
     cout<<"\n name is :"<< name;
     cout<<"\n department is :"<< dept;
}
```

```
};
class developer : protected employee
{
  string programminglanguage = "cpp";
  public :
  void display ()
{
  cout<<"\n emp id is : "<< empID
  cout<<"\n name is: "<< name;
  cout<<"\n programming language"<<programming
                                  language;

  }
};
int main ()
{
  manager m1;
  developer d1;
  m1. display ();
  d1. display ();
  return 0;
}
```

op

```
emp id is : 23
name is : bhushan
department name is : poly
emp id is : 23
name is : bhushan
programming language is : cpp
```

5) Hydrid inheritance-

combine multilevel and multiple inheritance
create a base class person with attributes
name and age. Derile class students from
person. Create 2 classes sports and academics
derile class results from students and
sports

code-

```
# include <iostream>
using name space std;
class person
{
    protected:
    string name = "bhushan";
    int age = 18;
};
class students : protected person
{
    protected:
    int academicsmark = 80,
};
class sports
{
    protected:
    int sports score = 50;
};
class academics
{
```

```
protected:
string grade = "A";
};
class result : protected student, protected
sports, protected academics
{
    int total score = sport score + academics
                      marks;
    public:
    void calculate ()
    {
    cout << " Total score is: << total score << "\n"
     grade is: " << grade;
    }
};
int main ()
{
    result r1;
    r1. calculate ();
    return 0;
}
```

Op -
total score is : 140
grade is : A

6) Virtual base class

code to implement following inheritance

class : college student
data mem student id

class : sports
data mem : grade

class : test
data mem : percentage

class : result.

code.

```
#include <iostream>
using namespace std;
class college student
{
  protected :
  int student_id;
  int college_code;
  public:
  void collectiondata ()
  {
  cout << " enter student id:";
  cin >> student_id;
  cout << " enter college code";
  cin >> college_code;
  }
  void display ()
  {
```

```
cout<<"\n student id:<<student_id;
cout<<\n college_code:<<college_code;
 }
};
class test : virtual public college student
{
    protected:
    float percentage;
    public:
    void set test data ()
    {
     cout<<"enter percentage:";
     cin>>percentage;
    }
    void display test data ()
    {
     cout<<"\n percentage:<<percentage;
    }
};
class sports : virtual public college student
{
    protected:
    char grade;
    public:
    void get sport data ()
    {
     cout<<"enter sport grade:";
     cin>>grade;
    }
```

```
void display sport data ()
{
 cout<<" in sport grade : <<grade;
}
};
class result : public test, public sports
{
    public:
    void getdata ()
    {
      collect data ();
      get test data ();
      get sports data ();
    }
    void displaydata ()
    {
      cout<<" result : \n";
      display ();
      display test data ();
      display sport data ();
    }
};
int main ()
{
    Result r;
    r. getdata ();
    r. display data ();
    return 0;
}
```

<grade,

ports

O/P -
enter student id: 1
enter college code: 34
enter percentage : 89
enter sports grade : A

result:
student id: 1
college code : 34
percentage : 88
sport grade : A

30/9/25

PAGE No.
DATE / /

Experiment →

1) WAP using function overloading to calculate the area of a laboratory which is rectangular in shape) and area of classroom which is square in shape.

code:

```
#include <iostream>
using namespace std;
class calculate area
{
  public:
  void area (int a, int b)
  {
    cout<<"\n area of lab is:"<< a*b;
  }
  void area (int s)
  {
    cout<<"\n area of class is << s*s;
  }
};
int main ()
{
  calculate area A;
  int a,b, side;
  cout << "enter a,b, side \n",
  cin >> a>>b >> side;
  A. area (a,b);
  A. area (side);
```

to calculate
cd is
of class-

a*b;

5*5;

return 0;
}

o/p -
enter a,b,side : 1,2,3
area of rect is:2
area of class is:9

2) WAP using function overloading to calculate sum of 5 float values and sum of 10 integers

code:

```
# include <iostream>
using namespace std;
class sumover
{
    int num, sum=0;
    float, fnum, fsum =0.0;
    public:
    void sumint (int)
    {
      for (int i=0; i<10; i++)
      {
        cout << "enter the integer";
        cin >> num;
        sum = sum +num;
      }
      cout << sum;
    }
    void sumint (int)
    {
      for (int i=0; i<10; i++)
      {
        cout << "enter the integer",
        cin >> num;
        sum = sum +num;
      }
```

```
cout<< sum << " \n";
}
void sumint (float)
{
    for (int i=0; i<5; i++)
    {
      cout<< " enter the float ";
      cin>> f num;
      f sum = f sum + f sum;
    }
    cout << f sum;
  }
};
int main ()
{
samover s;
s.sumint (0);
s.sumint (0.0f);
return 0;
}
```

O/p.

```
enter integer : 1
"          "  : 2
"          "  : 3
"          "  : 4
"          "  : 5
"          "  : 6
"          "  : 7
"          "  : 8
"          "  : 9
```

PAGE No.
DATE / /

55
enter Float : 3.3
enter Float : 4.4
enter Float : 4.4
enter Float : 4.4
enter Float : 2.1
18.0

17/10

2) WAP to implement unary - operator when used with the object so that the numeric data member of the class is negated.

code-

```
# include <iostream>
using namespace std;
class minus
{
  public:
  int A;
  void operator - ()
  {
      A = - A;
  }
};
int main ()
{
  minus B;
  B.A = 25;
  cout << "Before " << B.A << endl;
  -B;
  cout << "After" << B.A << endl;
  return 0;
}
```

O/P -
Before : 25
After : 24

4) WAP to implement the unary ++ operator when used with the object so that the numeric data member of the class is incremented.

code-

```
# include <iostream>
using namespace std;
class add
{
    public:
    int A;
    void operator ++()
    {
        A++;
    }
};
int main ()
{
    add B;
    B.A = 25;
    cout<<" Before ": << B.A <<endl;
    ++B;
    cout<<" After:"<< B.A << endl;
    return 0;
}
```

PAGE No.
DATE / /

O/P :
Before : 25
After : 26

17/10

Experiment - 8

1) WAP to overload the + operator so that the string's can concatenated.

code -

```
# include <iostream>
# include <cstring>
using namespace std;
class string
{
    private:
    char str [50];
    public:
    void accept ()
    {
      cout << " enter a string:";
      cin >> str;
    }
    void display ()
    {
      cout << str << endl;
    }
    void operator + (string s2)
    {
      strcat (this -> str, s2.str);
    }
};
```

```
int main ()
{
 string s1, s2;
 s1. accept ();
 s2. accept ();
 s1 +s2;
cout<< "First string is now: ";
 s1. display ();
cout<<" second string is still: ";
 s2. display ();
 return 0;
}
```

O/P-

```
Enter a string :abc
Enter a string : xyz
First string is now: abcxyz
second string is now: xyz
```

Experiment - 9

12 WAP to copy contents of one file to another

code -

```
#include <iostream>
#include <fstream>
using namespace std;
int main ()
{
 fstream first_file, second_file;
first_file open ("First.txt", ios::in);
 if (!first_file)
 {
  cout<<"error opening first file!";
  return 1;
 }
second_file open ("second.txt ios::out);
if (!second_file)
{
  cout<<"error opening second.txt!";
  return 1;
}
char ch;
while (first_file.get(ch))
{
  second_file.put(ch);
}
```

```
first_file.close();
second_file.close();
cout << "File copied successfully";
return 0;
}
```

2) WAP to count digit's and spaces

```
#include <iostream>
#include <fstream>
#include <ctype>
using namespace std;
int main ()
{
 fstream new_file;
new_file.open ("first.txt", ios::in);
 if (! new_file)
 {
    cout<<" error";
    return 1;
 }
 int dig_c =0
 int spc_c =0
 char ch;
 while (new_file.get(ch))
 {
   if (isdigit (ch))
   {
      dig_c++;
   }
   else if (isspace (ch))
   {
      spc_c++;
   }
 new_file.close();
 cout<<" No of digit :"<< dig_c;
```

```
cout<<" No of spaces"<< spc-c;
return 0;
}
```

3) WAP to count word's

```
#include <iostream>
#include <fstream>
#include <ctype>
using namespace std;
int main ()
{
  fstream new_file;
  new_file.open("First.txt", ios::in);
  if (!new file)
  {
    cout<<" error";
    return 1;
  }
  char ch;
  int word_c=0
  bool inword = false;
  while (new_file get(ch))
  {
    if (is space(ch))
    {
      inword = false
    }
    else if (! inword)
    {
      word_c ++;
      inword = true;
    }
```

```
new_file.close();
cout<<" No of word's";<<word
return 0;
}
```

4) Find occurance of given word

```
# include <iostream>
# include <fstream>
# include <string>
using namespace std;
int main()
{
 fstream new_file;
new_file.open("first.txt", ios::in);
if ( ! new_file)
{
   cout<<" error";
   return 1;
}
string target = "bhubban";
string word;
int count =0;
while ( new_file >>word)
{
    if (word = target
    {
        count ++;
    }
}
new_file close ();
cout<< "The word:"<< target<< "occurend"
<<count <<"times"<<endl;
   return 0;
}
```

Experiment -10

1) WAP to find sum of array elements using function template

```
#include <iostream>
using namespace std;
template <typename T>
T sumArray (T arr[])
{
        T sum =0;
        for(int i=0; i<10; i++)
        {
            sum += arr[i];
        }
        return sum;
}
int main ()
{
  int intarr[10]={1,2,3,4,5,6,7,8,9,10};
  float floatarr[10]={1.1F, 2.2F, 3.3F, 4.4F, 5.5F,
        6.6F, 7.7F, 8.8F, 9.9F, 10.0F};

  int intsum = sumArray (intarr);
  cout<<" sum of integer array :"<<intsum;
  float floatsum=sumArray (floatarr);
  cout<<" sum of float array :"<<floatsum;
  return 0;
}
```

PAGE No.
DATE / /

O/P.
Sum of integer array : 55
Sum of float array : 59.5

2) Prog to square function using template specilization

```
#include <iostream>
#include <string>
using namespace std;
template <typename T>
T square (T val)
{
    T sq = val * val;
    cout << sq;
    return sq;
}
template<>
string square (string squalue)
{
    string squaree = squalue + squalue;
    cout << squaree;
    return squaree;
}
int main ()
{
    int A = 5;
    string B = "abc";
    square (A);
    square (B);
    return 0;
}
```

PAGE NO.
DATE / /

O/P -
25
abc abc

3) WAP to built simple calculator using class template.

```
# include <iostream>
using namespace std;
template <class T> class
{ class c
{ public: T num1;
  num2;
  void accept ()
  {
   cout << "enter num1";
   cin >> num1;
   cout << "enter num2:";
   cin >> num2;
  }
  void add ()
  {
   cout << "Addition =" << num1 + num2;
  }
  void sub ()
  {
   cout << " subtraction:" << num1 - num2;
  }
  void mul ()
  {
   cout << "multiplication" << num1 * num2;
  }
```

```
void div ()
{
  if (num2 != 0)
  {
    cout<< "Division=" << num1/num2;
  }
  else
  {
  cout<< "Division by zero not possible";
  }
};
int main ()
{
  c <int> obj;
  obj .accept ();
  obj .add ();
  obj .sub ();
  obj .mul ();
  obj .div ();
  return 0;
}
```

o/p.

```
enter num1 =2
enter num2 =2
add = 4
sub =0
mul =4
Div =1
```

4) WAP to implement push and pop method from stack using class template.

```
#include <iostream>
using namespace std;
template <class T>
class stack
{
 T stack [100];
 int top;
 public:
 stack ()
 {
   top =-1;
 }
void push (T value)
{
 if (top= =99)
 {
  cout<<"stack overflow";
 }
 else
 {
  top ++;
  stack [top] = value;
  cout << value << "push into stack";
 }

}
```

```
void pop ()
{
  if (top ==-1)
  {
   cout<< " stack underflow";
  }
  else
  {
   cout<< stack [top]<< " popped from stack";
   top --;
  }
}
void display ()
{
  if ( top ==-1)
  {
   cout<< "stack is empty";
  }
  else
  {
   cout<< "stack elements:";
   for (int i= top; i>=0; i--)
    cout<< stack [i]<< " ";
    cout<< endl;
   }
  }
};
int main ()
{
   stack <int> s;
```

```
5. push
5. push
5. push
5. disp
5. pop
5. dis
return
}

O/P -
10 p
20 p
30 p
stack
30 p
stack
```

```
s.push(10);
s.push(20);
s.push(30);
s.display();
s.pop();
s.display();
return 0;
}
```

O/P:-

10 pushed
20 pushed
30 pushed
stack element : 30 20 10
30 popped
stack element : 20 10

Experiment -11

1) Accessing vector using iterator's

-> #include <iostream>
#include <bits/stdc++.h>
using namespace std;
int main ()
{
vector <char> v (10);
vector <char> ;
int i;
p=v.begin();
i=0;
while (p: =v.end()) {
* p= i+a;
p++;
i++; }
cout<< "original content:\n";
p = v.begin ();
while (p: = v.end()) {
cout<< * p<< " ";
p++;
}
}

O/P -
vector co
a b c

alor's

o/p-
vector content-

a b c d e f g h i j

2) without iterator-

```
-> #include <iostream>
   #include <bits/stdc++.h>
   using namespace std;
   int main ()
   {
     vector <char> v (10);
     unsigned int i;
     cout<<"size"=<< v.size();
     for( i=0; i<0; i++)
     {
        v[i] = i+a;
     }
     cout<<"current content:";
     for (i=0; i<v.size(); i++){
       cout<< v[i] << " ";
     }
     cout<<"\n \n";
     cout<<"expanding vector:";
     for( i=0; i<10; i++)
     {
        v.push_back( i+10+ 'a');
     }
     cout<<"size now="<<v.size();
     cout<<"current contents :\n";
     for (i=0; i<v.size(); i++)
     {
        v[i] = toupper [v[i]];
     }
```

```
cout<< "modified content's \n";
for (i=0; i<v.size (); i++)
{
  cout<< v[i] << " ";
}
return 0;
}
```

O/P.

content elements in the vector are:

a b c d e f g h i j

Expanding vector:

A B C D E F G H I J

PAGE No.
DATE / /

Experiment-12

1) Implement stack.

```
-> #include <bits/stdc++.h>
using namespace std;
{
    stack <char> car;
cars.push ("BMW");
cars.push ("Audi");
cars.push ("Mercedes");
cars.push ("Ferrari");
cout<< "Top element is:"<<cars.top();
cout<<size of stack is: <<cars.size();
cars.pop();
cars.pop();
while (! cars.empty())
{
    cout<< "element in stack are"<< cars.top();
    cars.pop();
}
return 0;
```

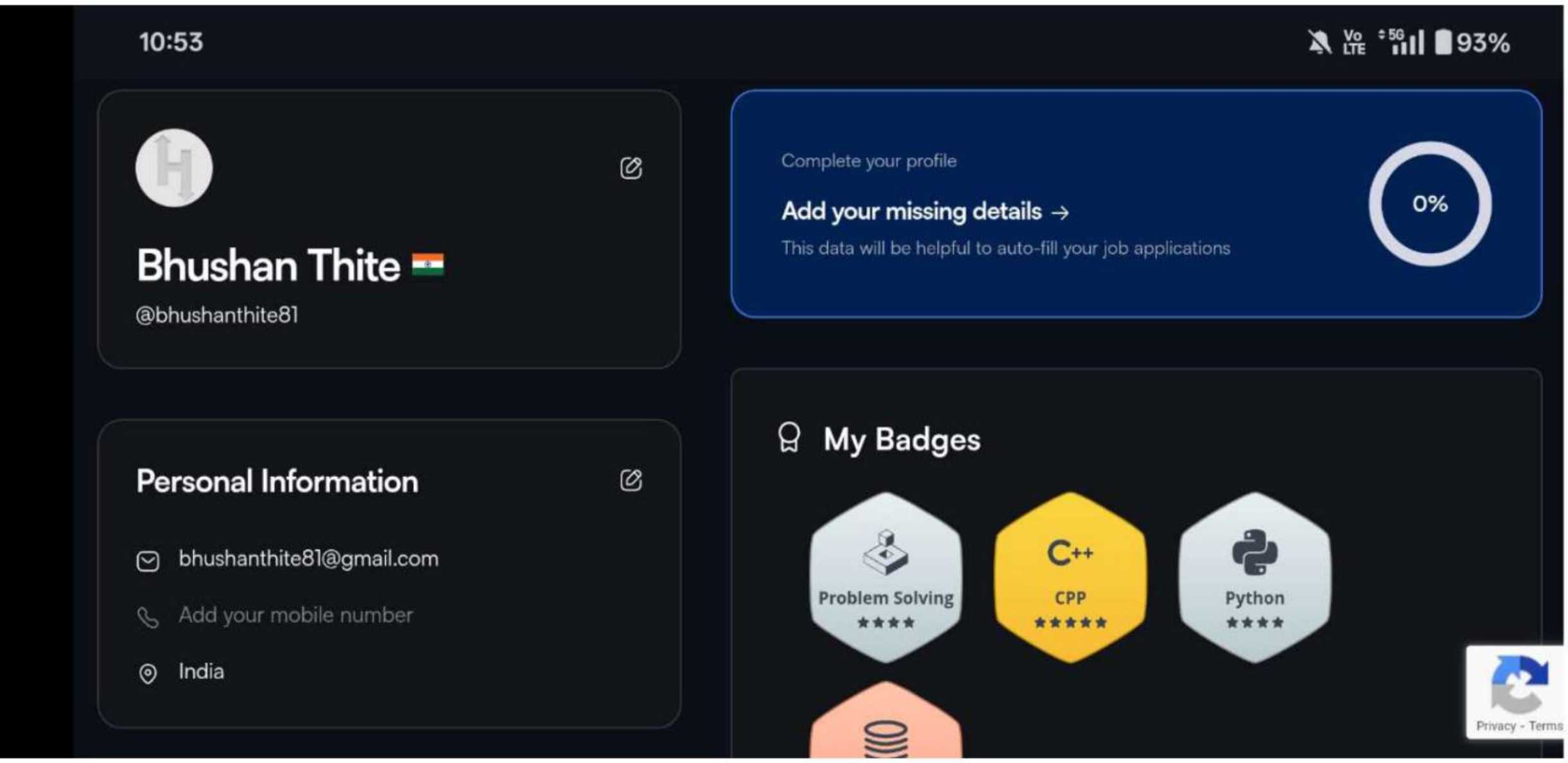
O/P -
Top element : Ferrari
size of element is: 4
Element stack are : Audi
: BMW

2) Implement queue

```
-> #include <bits/stdc++.h>
using namespace std;
int main ()
queue <int> age;
age.push(21);
age.push(22);
age.push(23);
age.push(24);
cout<< "Front element is:" << age.front();
cout<<" Back element is:"<< age.back();
age.pop();
age.pop();
while (! age.empty())
{
  cout << " element in the queue are:"<<
  age.front();
  age.pop;
}
return 0;
}
```

o/p:

Front element is: 21
Back element : 24
elements in the queue are : 23
Element in queue are: 24


10:53
93%
Bhushan Thite
@bhushanthite81
Complete your profile
Add your missing details →
This data will be helpful to auto-fill your job applications
0%
Personal Information
bhushanthite81@gmail.com
Add your mobile number
India
My Badges
Problem Solving
CPP
Python
Privacy - Terms