15/7/25

Date

Experiment - 1

1) WAP to declare a class student having data members as name, Roll no. Accept & display data for one student.

→

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

class student
{
int roll_no;
string name;
Public

void accept()
{
cout << "enter student name & roll_no:";
cin >> name >> roll_no;
}
void disp()
{
cout << "name of student:" << name;
cout << "Roll no. of student:" << roll no;
};
}; int main()
{
student s1;
s1.accept();
s1.disp();

return 0;
}
```

Output :

```
int main()
{
  student s1;
  s1.accept();
  s1.disp();

  return 0;
}
```

- Output :

```
Enter student name:   Aseeda
Enter roll no :   32

Student details:
   Name: Aseeda
   Roll no: 32.
```

2) Write a program to declare a class book having data members as id, name, price. Accept data for 2 books & display data of book having greater price.

→

```
#include <iostream>
using namespace std;
class book {
Public:
     int id;
     string name;
     float price;
```

```
void input (){
 cout<<"enter book id :";
 cin >> id;

 cin.ignore();

 cout << "enter book name:";
```

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

  class book
{
  public:
     int bp, bpgs;
string n;
  void accept()
  {
    cout <<"enter name & price:";
    cin >> n >>bp;
     cout << enter pages:";
     cin>>bpgs;
 }
  void display()
{
    cout <<"book name:" <<n <<endl;
    cout <<" book price:" <<bp<< endl;
    cout <<"book pages:" << bpgs <<endl;
}
};
```

```
int main ()
{
  book b1, b2;
  b1.accept() ;
  b2.accept();

if (b1.bp > b2.bp)
{
     cout <<"\n book with higher price:\n";
      b1.display();
 }
   else
{
   cout <<"\n book with higher price:\n";
   b2.display ();
}
   return 0;
}
```

- Output:

```
Enter name & price:  ABC  & 50
Enter pages        : 200
Enter name & price : XYZ  & 100
Enter pages        : 300

Book with higher price:

Book name = XYZ
Book price = 100
Book pages = 300.
```

3) WAP to declare a class time having data members as H, m & S. Accept data for one object & display total time in seconds.

→
```
#inclued<iostream>
#inclued

#include <iostream>
#include <string>

using namespace std;
class time
{
  int h, m, s, t;
public:
  void accept()
{
  cout << "enter hours:";
  cin >> h;
  cout << "enter minutes:";
  cin >> m;
  cout << "enter seconds:";
  cin >> s;
 }
  void display()
 {
  t = (h * 3600) + (m * 60) + s
   cout << "total time:" << t << endl;
 }
};
```

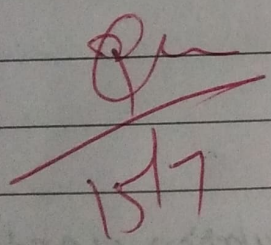
```
int main()
{
  time t1;
  t1.accept();
  t1.display();

  return 0;
}
```

- output:

```
Enter hours: 4
Enter minutes: 6
Enter seconds: 89

total time= 14849.
```



Experiment - 2

1 WAP to declare class 'city' (~~student~~) having data members as ~~roll-no~~ & name & ~~class~~ & population. Accept this data for 5 cities & display name of city having highest population.

⇒

```
#include <iostream>
using namespace std;
class city {
 Public:
 string name;
 int population;
 void accept(){
 cout<< "enter city name:";
 cin>> name;
 cout << "enter population:";
 cin >> population;
 }
  void display(){
 cout<<"city:"<<name <<"population:"<<population <<endl;
}
};
 int main(){
 city cities[5];
 for(int i=0; i<5; i++){
 cout <<"\n enter information of cities:"<<i+1<<"\n";
 cities[i].accept();
}
int max=0;
for(int i=1; i<5; i++){
if(cities[i].population> cities[max].population){
 max=i;
```

```
}
}
cout<<"\n city with highest population :\n";
cities [max].display();
   return 0;
}
```

23 ■ output :

- Enter information of cities 1 :
  - enter city name: sde
    enter population: 21
- Enter information of cities 2:
  enter city name: gdf
  enter population: 45.
- Enter information of cities 3:
  enter city name: iyhg
  enter population: 68
- Enter information of cities 4:
  enter city name: hgf
  enter population: 67
- enter information of cities 5:
  enter city name: gdser
  enter population: 88.
- city with highest population:
  city: gdser
  population: 88.

2) Write a program to declare a class 'account' having data members as account no. & balance. Accept this data for 10 account & give interest of 10% where balance is = or greater than 5000 & display them.

→

```
#include <iostream>
using namespace std;
class account {
int accno;
float balance;
Public:
void accept() {
cout<<"enter account number:";
cin >> accno;
cout<<" enter balance:";
cin >> balance;
}
void addinterest (){
if (balance >=5000) {
balance += balance * 0.10;
}
}
void display (){
if (balance >=5000){
cout<< "account no. :"<< accno<< "balance with
interest:" << balance << endl;
}
}
};
int main() {
```

```
account acc [10];
for (int i=0; i<10; i++){
cout << "\n enter details for account" << i+1 << ":\n";
acc[i].accept();
acc [i].disaddinterest();
}
cout << "\n accounts with balance >=5000 after adding
  interest:\n";
for(int i=0; i<10; i++){
   acc [i].display();
}
   return 0;
}
```

■ Output:

- enter balai details for account 1:
  enter account number: 12
  enter balance: 345

- enter details for account 2:
  enter account number: 56
  enter balance: 876

- enter details for account 3:
  enter account number: 45
  enter balance: 897

- enter details for account 4: 334
  enter account number: 334
  enter balance: 00897

- enter details for account 5:
  enter account number: 55
  enter balance: 678

3)

=>

- enter details for account 6:
  enter account number: 666
  enter balance: 7788

- enter details for account 7:
  enter account number: 78
  enter balance: 8888

- enter details for account 8:
  enter account number: 45
  enter balance: 7890

- enter details for account 9:
  enter account number: 33
  enter balance: 5644

- enter details for account 10:
  enter account number: 33
  enter balance: 67865

- Accounts with balance >=5000 after adding interest:

account no: 666 balance with interest: 8566.8
account no: 78 balance with interest: 9776.8
account no: 45 balance with interest: 8679
account no: 33 balance with interest: 6208.4
account no: 33 balance with interest: 74851.5

Page No.
Date

3) Write a program to declare a class 'staff' having data members as name & post. Accept this data for 5 staff & display names of staff who are 'HOD'.

=>

```
#include <iostream>
#include <string>
using namespace std;
class staff{
string name;
string post;
public:
void accept(){
cout<<"enter name:";
cin>>post;
cout<<"enter post:";
cin>>post;
}
void display(){
if(post=="HOD"){
cout<<"name:"<<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 staff with post HOD:\n";
for(int i=0; i<5; i++){
```

```
    s[i].display();
  }
  return 0;
}
```

```
Enter the details for staff 1
        Enter staff Name: Juan
        Eenter staff Post : HOD

Enter details for staff 2
        Enter staff name : Maria
        Enter staff Post : Head

Enter the details for staff 3
        Enter staff name : Jane
        Enter staff Post : HOD

Enter details for staff 4
        Enter staff name : Mark
        Enter staff Post : HR

Enter details for staff 5
        Enter staff name: JHB
        Enter staff Post : Head

Staff who are HOD:
        Name of staff : Juan
        Name of staff : Jane
```

29/7/25

Experiment - 3

1) WAP to declare a class 'book' containing data members as book-title, author-name & price. Accept & display the info for one object using a pointer to that object.

→
```
#include <iostream>
using namespace std;
class book
{
  public:
      string book-title;
      int price;
      string author_name;

  void accept()
  {
    cout << "enter book name:";
    cin >> book_title;
    cout << "enter the name of author:";
    cin >> author_name;
    cout << "price of the book:";
    cin >> price;
  }
 void display()
{
  cout << "book name is:" << book_title;
  cout << "\n author name:" << author_name;
  cout << "\n price is:" << price;
 }
};
```

```
int main()
{
    book b1;
    book *p;
    b1.accept();
    b1.display();

    return 0;
}
```

book b1;
book *b1;
b1 → accept();
b1 → display();

- Output :

```
Enter book name    : Python
Enter name of author: Rusk
Price of the book : 999
  Book name is    : Python
  Author name is  : Rusk
    Price is      : 999.
```

2) WAP to declare a class 'student' having data members roll_no & %. Using 'this' pointer invoke member functions to accept & display this data for one object of the class.

->
```
#include <iostream>
using namespace std;
class student
{
 int roll_no;
 float percentage;
 public:
```

```
void accept()
{
cout<<"enter studen roll-no:";
cin>>this->roll_no;
cout<<"enter student percentage :";
cin>>this->percentage;
}
  void display()
{
cout<<"Roll number of student is:"<<roll_no;
cout<<"\n percentage is:"<<percentage;
}
};
 int main()
{
    student s1;
    student *p;
    p=&s1;
  s1.accept();
  s1.display();

    return 0;
}
```

■ OUTPUT :

```
Enter the student roll_number : 39
Enter the student percentage : 7.93
Roll number of studen is = 39
 percentage is = 7.93
```

3) WAP to demonstrate use of nested class.

→
```
#include<iostream>
using namespace std;
class number
{
 Private:
    int num;
    class operations
    {
    Public:
     int square (int n)
     {
      return n*n;
     }
     int cube(int n)
     {
      return n*n*n;
     }
};
    public:
   void accept()
{
 cout<<" enter an number:";
   cin>>num;
}
 void display()
{
 operations op;
```

```
cout<<" square of :"<<" ="<< op.square(num)<<endl;
cout<<" cube of :"<<"<<" ="<< op.cube(num)<<endl;
}
};
 int main()
{
 Number n1;
 n1.accept();
 n1.display();
 return 0;
}
```

- Output:

```
Enter an number : 2
square of : 4
cube of : 8
```

24/7/25

EXPERIMENT - 4

Q.1 WAP to swap two numbers from same class using object as function argument. Write swap functions as member function.

→

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

class number {
    int a, b;
Public:
    void input() {
    cout << "enter two numbers:";
    cin >> a >>b;
    }
    void display() {
    cout<<" a:" <<a<<" b :" <<b<< endl;
    }

    void swapnumbers (Number & obj) {
    int temp = obj.a ;
    obj.a = obj.b ;
    obj.b = temp ;
    }
};
int main() {
 number n1;
 n1. input();
 cout << " Before swapping :";
 n1.display();
 n1. swapnumbers (n1) ;
cout<<" After swapping:" ;
```

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

- Output :

Enter two numbers : 2 4

Before swapping : a = 2, b = 4
After swapping : a = 4, b = 2.

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

→

```
#include <iostream>
using namespace std;
class Number {
  int a, b;
Public:
  void input() {
   cout << "enter two numbers :";
   cin >> a >> b;
  }
  void display() {
   cout << " a:" << a << "b:" << b << endl;
  }
friend void swapnumbers (number & obj);
};
void swapnumbers (Number & obj) {
 int temp = obj.a;
```

```
    obj.a = obj.b;
    obj.b = temp;
}
int main() {
    Number n1;
    n1.input();
    cout << "before swapping:";
    n1.display();
    swapnumbers(n1);
    cout << "After swapping:";
    n1.display();
    return 0;
}
```

- Output :

Enter two numbers : 1 3

Before swapping :

a : 1 , b : 3

After swapping :

a : 3 , b : 1

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

->
```
# include <iostream>
using namespace std;
class class1;
class class2 {
int b;
Public:
    void input() {
    cout<<" enter number for class 2:";
     cin>>b;
    }
    void display() {
     cout <<" class b =" <<b<< endl;
    }
    friend void swapnumbers (class1 &x, class &y);
    };
    class class1 {
     int a;
  Public:
     void input(){
     cout<<" enter number for class1:";
     cin>>a;
   }
   void display () {
   cout<<" class1 a =" << a<< end1;
   }
   friend void swapnumbers (class1 &x, class &y);
   };
```

```
void swapnumbers ( class1 &x , class2 &y)
{
   int temp = x.a;
   x.a = y.b;
   y.b = temp;
}
int main() {
   class1 obj1;
   class2 obj2;

   obj1.input();
   obj2.input();

   cout<<" Before swapping:" <<endl;
   obj1.display();
   obj2.display();

   swapnumbers (obj1, obj2);

   cout<<" After swapping:"<<endl;
   obj1.display();
   obj2.display();

   return 0;
}
```

Page No.
Date

- Output:

```
Enter number for class 1 : 5
Enter number for class 2 : 2

Before swapping:
                class1 a = 5
                class2 b = 2

After swapping :
                class1 a = 2
                class2 b = 5.
```

4. WAP to create two classes Result1 & Result2 which stores the marks of the students. Read the value of a mars for both class objects & compute the average of two results.

→

```
#include <stdiciostream>
using namespace std;
class Results2;

class Results1 {
    int marks;
Public:
    void input(){
    cout << "enter marks for Results:";
    cin >> marks;
    }
    friend void average (Result r1, Result r2);
};
```

```
class results2 {
  int marks;
Public:
  void input () {
    cout<<" enter marks for results 2:";
    cin >> marks;
    }
  friend void average (result r1, result r2);
};

void average (result r1, result r2) {
     float avg = (r1.marks + r2.marks) / 2.0;
     cout<<" Avg marks =" << Avg << endl;
   }
int main() {
   Result 1 obj1;
   Result 2 obj2;

   obj1.input();
   obj2.input();

   average (obj1, obj2);
    return 0;
}.
```

- Output :

```
Enter marks for result 1 : 80
Enter marks for result 2: 90

Avg marks = 85
```