

28/9/2021

## Input and Output operations

Exp :-

Aim :- To find the total number of hrs and minutes.

Algorithm :-

Step 1: Start

Step 2: Declare the Variable

Step 3: take the value for the variables using cin

Step 4: Find the total number of hrs and min.

Step 5: By the total minutes divide by The constant minutes  
you can find hrs and by total minutes by percentage  
of the constant Minutes

Step 6: Print the desired output using cout

Step 7: Stop

Code:-

```
#include <iostream>
using namespace std;
int main()
{
    int tot_mins, hrs, mins;
    const int Minutes = 60;
    cin >> tot_mins;
    hrs = (tot_mins / Minutes);
    mins = (tot_mins % Minutes);
    cout << hrs << " Hours and " << mins << " Minutes ";
    return 0;
}
```

### Output :-

i) 546

9 Hours and 6 minutes

ii) 101

1 Hours and 41 minutes

Algorithm:

Step 1: take the values for the variables using cin

Step 2: Find the difference between the two driving speed

Step 3: Using if else compare the result

Step 4: Print the desired output using cout

Step 5: Stop

Code:-

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

### Result:-

Thus the given program is executed with the desired output.

```
speeddiff = accspeed - anspeed;
cout << speeddiff; }
```

return 0; }

## Exp<sup>2</sup>

Aim:- To find the difference between the driving speed of two participants.

Algorithm:-

- Step 1 : Start
- Step 2 : take the values for the variables using cin
- Step 3 : Find the difference between the two driving speed
- Step 4 : Using if else compare the result
- Step 5 : Print the desired output using cout.
- Step 6 : Stop

Code :-

```
#include <iostream>
using namespace std;
int main ()
{
    int aronspeed , aaronspeed , speeddiff ;
    cin >> aronspeed >> aaronspeed ;
    if (aronspeed > aaronspeed)
    {
        speeddiff = aaronspeed - aronspeed ; cout << speeddiff ;
    }
    else
    {
        speeddiff = aronspeed - aaronspeed ;
        cout << speeddiff ;
    }
    return 0 ;
}
```

## Output

i) 74  
51  
23

ii)  
76  
89  
13

Sum of all the angles should be equal to 180° of the triangle.

Algorithm:-

Step 1: Start

Step 2: Declare the variable

Step 3: take the value for the variables using cin  
Step 4: add all the angles

Step 5: Using if else compare the result.

Step 6: Print the desired output using cout

Result:-

Thus the given program is executed with the desired output.

```
int angle1, angle2, angle3, sumofangle;
cout >> angle1 >> angle2 >> angle3;
sumofangle = angle1 + angle2 + angle3;
if ((sumofangle == 180))
    cout << "Angles are valid";
else
    cout << "Angle are not valid";
return 0;
```

Exp 3:-

Aim :- Sum of all three angles should be equal to  $180^\circ$  of the triangle.

Algorithm :-

Step 1: Start

Step 2: Declare the variable

Step 3: take the value for the variables using cin

Step 4: add all the angles

Step 5: Using if else compare the result.

Step 6: Print the desired output using cout

Step 7: Stop

Code :-

```
#include <iostream>
using namespace std;
int main()
{
    int angle1, angle2, angle3, sumofangle;
    cin >> angle1 >> angle2 >> angle3;
    sumofangle = angle1 + angle2 + angle3;
    if (sumofangle == 180)
        cout << "Angles are valid";
    else
        cout << "Angle are not valid";
    return 0;
}
```

Output :-

i) 90  
55  
35

Angle are valid

ii) 67  
49  
54

Angle are not valid

(8)

Result :-

Thus the given program is executed with the desired output.

## Expt - 1

### Classes, Method & constructor

aim: to convert a standard phone number to a updated phone number.

algorithm:

Step 1: Start

Step 2: Accept variable float, height, name, weight;

Step 3: apply wid display data

Step 4: using cout find the output

Step 5: Stop.

Code :

```
#include <iostream>
#include <string.h>
using namespace std;
class student
{
public:
    string name;
    int roll;
    float name, height, weight;
    student() {name = "Bhagavan"; roll = 1593; height = 172.5;
    weight = 60.4; }
```

```
void display data()
```

```
{ cin >> name;  
cin >> roll;  
cin >> height;  
cin >> weight;  
cout << name << " ";  
cout << roll << " ";  
cout << height << " ";  
cout << weight << endl;
```

```
};
```

```
int main()
```

```
{
```

```
student s1, s2;
```

```
s1.display data();
```

```
s2.display data();
```

```
return 0; }
```

Output

i) Maheshwaran 157 162.1 68.1      ii) Vinoth 186 - 171.5 53.3  
Maheshwaran 157 162.1 68.1  
Vinoth 186 171.5 53.3  
Bhagavan 1593 172.5 60.4  
Bhagavan 1593 172.5 60.4

Result:

The above programs have been compiled and executed successfully.

## BPT-2

Aim:

To count the possible ways for counting drinks

Algorithm:

- Step 1: Start
- Step 2: accept variable  $n, a, b, c, i, j, rem$  and
- Step 3:  $rem = n - i - j - j;$
- Step 4:  $ans += (rem > 0 \& rem \leq 2c = a);$
- Step 5: End

Code:

```
#include <iostream>
using namespace std;
int n, a, b, c, i, j, rem, ans;
int main() {
    cin >> n >> a >> b >> c;
    if (n == 0)
        cout << "class drinks public :: void shop () By .shop();"
    for (i=0; i<=b; i++) {
        for (j=0; j<=c; j++) {
            rem = n - i - j - j;
            ans += (rem > 0 & rem <= 2c = a);
        }
    }
    cout << ans;
}
```

```
return;
```

```
}
```

### Output

```
25 10 5 10  
12
```

### Code:

```
#include <iostream>  
using namespace std;  
class student{
```

```
public:
```

```
void name(string s1, string s2, string s3){
```

```
string s1 = "Rahul";
```

### Result:

The above programs have been compiled and executed successfully

```
33  
int main()
```

```
{
```

### Expt - 3

Aim:

To create a unique verification for validating aadhar using classes

Algorithm:

Step 1: Start

Step 2: apply variable class aadhar

Step 3: if first name, last name, middle name present.

Step 4: then valid else invalid

Step 5: Stop

Code:

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

class Aadhaar {

public:

void nameOfCitizen (string fn, string mn, string ln) {

string str = (ln != " ") & (fn != " ") &

(mn != " ") ? fn + mn + ln : "invalid Name";

cout << str;

3

};

int main()

{

String fn, mn, ln;  
aadhaar card;  
fn >> fn >> mn >> ln;  
card. Name of citizen (fn, mn, ln);  
return 0;

3

Output

Vidhya  
Pradeep  
Invalid name

Result: 18

The above programmes have been compiled and executed successfully.

## Game shop UML

Register onto the website



Consumer

Login to the website

Browse item

Search item

View item description

View recommended items

Order and payment

Customer zone

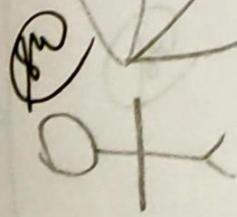
Sell item

Manage products

Maintain the website

Phone call

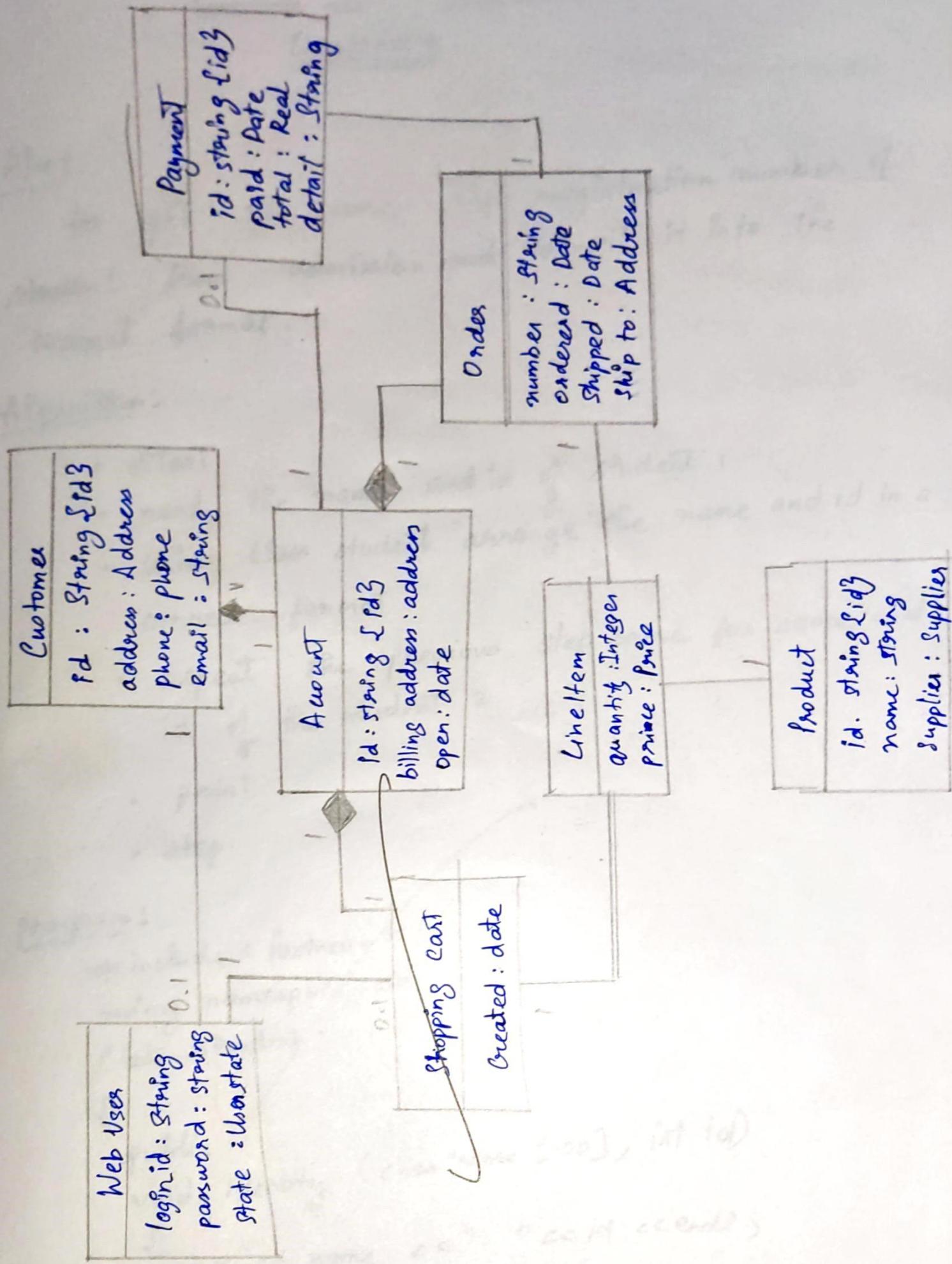
Website



Seller

E-commerce  
admin

## Class game shop



## Function and Constructor

### Overloading

#### Aim:

to get the name and registration number of student from admission and convert it into the correct format.

#### Algorithm:

- start
- read the name and id of student
- Using class student arrange the name and id in a correct format
- repeat the previous step same for name and id of the student 2
- print
- stop

#### Program:

```
#include <iostream>
using namespace std;
class student {
public:
    void identity (char name [100], int id)
    {
        cout << name << " " << id << endl;
    }
    void identity (int id, char name [10])
    {
        cout << name << " " << id << endl;
    }
};
```

int main()

{

char name[20];

Output:

1) Harsh

1930405078

1930405079

Amrit

Harsh

1930405078

Amrit

1930405079

```
int main()
{
    char name[100];
    int id;
    cin >> name >> id;
    student details;
    details.Identity(name, id);
    cout >> od >> name;
    detail.Identity(id, name);
    return 0;
}
```

Result:  
The program has been compiled and executed  
successfully

2) Aim:  
to calculate the Bill for hospital rooms and medicines every week

Algorithm:

- Start
- Read the amount of medicines
- Read the amount of room and no. of days.
- Using Function overloading calculate the Bill for room and medicines
- Stop

Program:

```
#include <iostream>
using namespace std;
class hospital
{
public:
    float bill (float medicine , float days)
    {
        float sum = medicines * days ;
        return sum ;
    }
    int bill (int room , int days)
    {
        int sum = room * days
        return sum ;
    }
}
```

## Output

D 5000  
7 15000  
7

→ 350000  
105000

Taste

Brown salt taste

Yellow salt taste

MS salt stimulus produces neither taste

minimize bias more nof

get

presenting stimulus #  
LBT2 response time  
latency (ms)

(spont taste + aversive taste) MS taste

spont + aversive = max taste

max ranking

(spont tri + max tri) MS tri

spont + max = max tri

max rating

```
int main ()  
{  
    Hospital ob;  
    float medicinebill , day ;  
    cin >> medicinebill >> days ;  
    ob . bill (medicinebill , days ) ;  
    cout << ob . bill (medicinebill , days ) << endl ;  
  
    float roomrent  
    cin >> roomrent >> days ;  
    ob . bill (roomrent , days ) ;  
    cout << ob . bill (roomrent , days ) << endl ;  
  
    return 0 ;  
}
```

### Result:

The program has been compiled and executed successfully

3)

### Aim :

To show "welcome" followed by "First name and last name" when the user login into his account.

### Algorithm :

- Start
- Read the first name and Last name
- Print Hi before First name and Welcome before first name and Last name.
- Print
- Stop

### Program :

```
#include <iostream>
using namespace std;
class Welcoming
{
public:
    int msg (char fname [100])
    {
        cout << "Hi" << fname << endl;
        return 0;
    }
    int msg (char fname [100], char name [100])
    {
        cout << "Welcome" << fname << " " << name << endl;
    }
}
```

## Output:

Mahendra

Sing<sup>h</sup>

Mahendra

Sing<sup>h</sup>

Dhoni

→ Hi Mahendra Sing<sup>h</sup>

Welcome

Mahendra Sing<sup>h</sup> Dhoni

J

connection → student life  
is being made → final  
examination going

((00) work not )

((00) work not ) gain not

((00) work not ) "H" is true

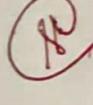
(o nature

((00) work not , (00) work not ) gain not

→ work not " work not " not " work " is true

(work

```
return 0;  
}  
};  
int main ()  
{  
    Welcome msg ob;  
    char fname [100], lname [100];  
    cin >> fname >> lname;  
    ob.msg (fname);  
    ob.msg (lname);  
    return 0;  
}
```

Result:  
 The program has been compiled and  
executed successfully.