

**Question - 1**

Write a complete C++ program to do the following :

- (i) 'Student' is a base class, having two data members: entryno and name; entryno is integer and name of 20 characters long. The value of entryno is 1 for Science student and 2 for Arts student, otherwise it is an error.
- (ii) 'Science' and 'Arts' are two derived classes, having respectively data items marks for Physics, Chemistry, Mathematics and marks for English, History, Economics.
- (iii) Read appropriate data from the screen for 3 science and 2 arts students.
- (iv) Display entryno, name, marks for science students first and then for arts students.

**CODE-**

Terminal

```
File Edit View Search Terminal Help
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5
6 class student {
7 protected:
8     int entryno;
9     char name[20] ;
10
11 public:
12     void getstud() {
13         cout << " enter entry number 1 for science, 2 for arts" << endl;
14         cin >> entryno;
15
16         if (entryno != 1 && entryno != 2) {
17             cout << "invalid number" << endl;
18             exit(0);
19         }
20         cout << "enter name";
21         cin >> name;
22     }
23
24     void putstudent() const
25     {
26         cout << "enter_num" << " " << entryno << " << "name" << " " << name << endl;
27     }
28 };
29
30 class science : public student {
31     int phy, chem, math;
32
33 public:
34     void get_data()
35     {
36         getstud();
37         cout << "enter physics marks-";
38         cin >> phy;
39         cout << "enter chemistry marks-";
40         cin >> chem;
41         cout << "enter maths marks-";
42         cin >> math;
43     }
44
45     void putData() const
46     {
47         putstudent();
48         cout << "physics " << phy << " chemistry " << chem << " maths " << math << endl;
49     }
50 };
51
```

```

53 class arts : public student {
54     int eng, history, eco;
55
56 public:
57     void get_data() {
58         getstud();
59         cout << "english marks- ";
60         cin >> eng;
61         cout << "history marks- ";
62         cin >> history;
63         cout << "economics marks ";
64         cin >> eco;
65     }
66
67     void putdata() const {
68         putstudent();
69         cout << "english " << eng << " history " << history << " economics " << eco << endl;
70     }
71 };
72
73 int main() {
74     science s[3];
75     arts a[2];
76
77     cout << "science student details" << endl;
78     for (int i = 0; i < 3; i++) {
79         cout << endl << "science student" << i + 1 << endl;
80         s[i].get_data();
81     }
82
83     cout << "art student info" << endl;
84     for (int i = 0; i < 2; i++) {
85         cout << "art student" << i + 1 << endl;
86         a[i].get_data();
87     }
88
89     cout << "science student";
90     for (int i = 0; i < 3; i++) {
91         cout << endl << "student" << i + 1 << endl;
92         s[i].putData();
93     }
94
95     cout << "art students" << endl;
96     for (int i = 0; i < 2; i++) {
97         cout << endl << "student" << i + 1 << endl;
98         a[i].putdata();
99     }
100
101     return 0;
102 }
```

## Question – 2

An electricity board charges the following rates to domestic users to discourage large consumption of

energy :

For the first 100 units –50 P per unit

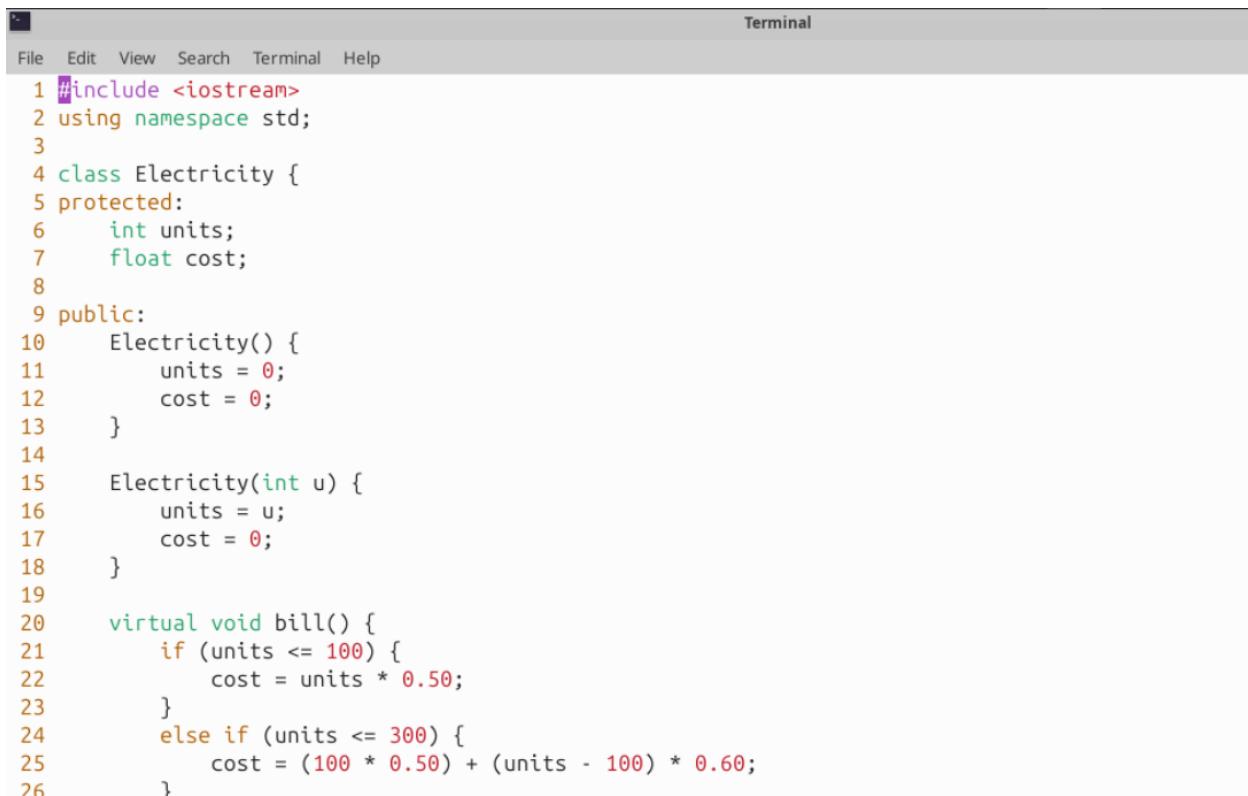
Beyond 300 units– 60 P per unit

If the total cost is more than Rs.250.00 then an additional surcharge of 15% is added on the difference.

Define a class Electricity in which the function Bill computes the cost. Define a derived class

More\_Electricity and override Bill to add the surcharge

CODE-



The image shows a terminal window with the title "Terminal". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The code is displayed in the terminal window:

```
File Edit View Search Terminal Help
1 #include <iostream>
2 using namespace std;
3
4 class Electricity {
5 protected:
6     int units;
7     float cost;
8
9 public:
10    Electricity() {
11        units = 0;
12        cost = 0;
13    }
14
15    Electricity(int u) {
16        units = u;
17        cost = 0;
18    }
19
20    virtual void bill() {
21        if (units <= 100) {
22            cost = units * 0.50;
23        }
24        else if (units <= 300) {
25            cost = (100 * 0.50) + (units - 100) * 0.60;
26        }
}
```

```

26     }
27     else {
28         cost = (100 * 0.50) + (200 * 0.60) + (units - 300) * 0.80;
29     }
30
31     cout << "Bill without surcharge rs-" << cost << endl;
32 }
33 };
34
35 class More_electricity : public Electricity {
36 public:
37     More_electricity(int u) : Electricity(u) {}
38
39     void bill() override {
40         Electricity::bill();
41
42         if (cost > 250) {
43             float surcharge = (cost - 250) * 0.15;
44             cost += surcharge;
45         }
46
47         cout << "final bill with surcharge rs" << cost << endl;
48     }
49 };
50
51 int main() {
52     int ut;
53     cout << "units ";
54     cin >> ut;
55
56     More_electricity s1(ut);
57     s1.bill();
58
59     return 0;
60 }
```

### Question – 3

Create a class Time that has separate int member data for hours, minutes and seconds.

One constructor

should initialize this data to zero and another constructor initialize it to fixed values. Write member

function to display time in 12 hour as well as 24 hour format. The final member function should add

two objects of class Time.

A main() program should create three objects of class time, of which two are initialized to specific

values and third object initialized to zero. Then it should add the two initialized values together,

leaving the result in the third. Finally it should display the value of all three objects with appropriate headings.

CODE-

Terminal

```
File Edit View Search Terminal Help
1 #include <iostream>
2 using namespace std;
3
4 class time {
5     int hrs, min, sec;
6
7 public:
8
9     time()    {
10         hrs = 0;
11         min = 0;
12         sec = 0;
13     }
14
15
16
17     time(int h, int m, int s)
18     {
19         hrs = h;
20         min = m;
21         sec = s;
22     }
23
24
25     void display_24() const {
26
27         cout << hrs << ":" << min << ":" << sec << endl;
28     }
29
30
```

```
30
31     void display_12() const    {
32         int h = hrs % 12;
33         if (h == 0)
34         {
35             h = 12;
36         }
37
38         cout << h << ":" << min << ":" << sec << endl;
39     }
40
41
42     time operator+(const time& t)  {
43         time temp;
44
45         temp.sec = sec + t.sec;
46
47         temp.min = min + t.min + temp.sec / 60;
48         temp.sec = temp.sec % 60;
49
50         temp.hrs = hrs + t.hrs + temp.min / 60;
51         temp.min = temp.min % 60;
52         temp.hrs = temp.hrs % 24;
53
54         return temp;
55
56
57     }
58 };
59
```

```

60 int main()
61 {
62
63     time t1(10, 43, 30);
64     time t2(5, 30, 15);
65     time t3;
66     t3 = t1 + t2;
67
68     cout << "time 1 ,24 hour";
69     t1.display_24();
70     cout << "time 1 ,12 hour";
71     t1.display_12();
72
73     cout << endl;
74
75     cout << "time 2 ,24 hour ";
76     t2.display_24();
77     cout << "time 2 ,12 hour ";
78     t2.display_12();
79
80     cout << endl;
81
82     cout << " answer time ,24 hour: ";
83     t3.display_24();
84     cout << "answertime ,12 hour ";
85     t3.display_12();
86
87     return 0;
88 }
```

#### Question – 4

Create a class Patient that stores the patient name (a string) and the disease (a string) of the patient.

From this class derive two classes : In\_patient which has a data member room\_rent (type float) and

Out\_patient which has a data member OPD\_charges (float). Each of these three classes should have a

nondefault constructor and a putdata() function to display its data. Write a main() program to test In\_

patient and Out\_patient classes by creating instances of them and then displaying the data with

putdata()

CODE-

## Terminal

File Edit View Search Terminal Help

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5
6 class patient {
7     protected:
8     char name[50];
9     char disease[50];
10
11     public:
12
13     patient(char* n, char* d) {
14         strcpy(name, n);
15         strcpy(disease, d);
16     }
17
18     void putdata() {
19         cout << " patient name " << name << endl;
20         cout << "disease name " << disease << endl;
21     }
22 };
23
24 class In_patient : public patient {
25     float roomrent;
26 }
```

```

27 public:
28     In_patient(char* n, char* d, float rent): patient(n, d) {
29         roomrent = rent;
30     }
31
32     void putdata()
33     {
34         patient::putdata();
35         cout << "rent is " << roomrent << endl;
36     }
37 }
38 };
39
40
41 class out_patient : public patient {
42     private:
43     float OPD_charge;
44
45     public:
46
47     out_patient(char* n, char* d, float charge) : patient(n, d) {
48         OPD_charge = charge;
49     }
50
51     void putdata() {
52         patient::putdata();
53         cout << "opd Charges " << OPD_charge << endl;
54     }
55 }
56 };
57
58 int main() {
59     In_patient s1("karn", "fever", 2400.5);
60     out_patient s2("sush", "cold", 500.9);
61
62     cout << "patient_info";
63     s1.putdata();
64
65     cout << "patients info";
66     s2.putdata();
67
68     return 0;
69 }

```

### Question – 5

Define a class Publication which has a title. Derive two classes from it – a class Book which has an accession number and a class Magazine which has volume number. With these two as bases, derive the

class Journal. Define a function print() in each of these classes. Ensure that the derived class function always invokes the base(s) class function. In main() create a Journal called IEEEOOP with an accession number 681.3 and a volume number 1. Invoke the print() function for this object.

## CODE-

```
File Edit View Search Terminal Help Terminal
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 class Publication {
6
7     protected:
8
9     char title[40];
10
11    public:
12
13    Publication(char* t) {
14        strcpy(title,t);
15
16    }
17
18    void print() {
19        cout << "title " << title << endl;
20
21    }
22
23 };
24
25
26 class Book : virtual public Publication {
27     protected:
28     float accession_no;
29
30     public:
31
32     Book(char *t, float a) : Publication(t)
33     {
34         accession_no = a;
35     }
36
37     void print() {
38         Publication::print();
39         cout << "access number -" << accession_no << endl;
40     }
41 };
42
43
44 class Magazine : virtual public Publication {
45
46     protected:
47
48     int volume_no;
```

```

50     public:
51
52     Magazine(char* t, int v) : Publication(t) {
53         volume_no = v;
54     }
55
56     void print() {
57
58         Publication::print();
59         cout << "vol. number-" << volume_no << endl;
60     }
61 }
62
63 };
64
65 class Journal : public Book, public Magazine {
66
67     public:
68
69     Journal(char* t, float a, int v) : Publication(t), Book(t, a), Magazine(t, v)
70     {
71
72     }
73
74     void print()
75     {
76         Publication::print();
77         cout << "acces_num -" << accession_no << " " << "vol_num=-" << " " << volume_no << endl;;
78     }
79 }
80 };
81
82 int main() {
83     Journal s1("IEEOOP", 681.3, 1);
84
85     cout << "Journal" << endl;
86     s1.print();
87
88     return 0;
89 }
```

### Question – 6

Assume that the cell users are two kinds – those with a post paid option and those with a prepaid

option. Post paid gives a fixed free talk time and the rest is computed at the rate of Rs.1.90 per pulse.

Prepaid cards have a fixed talk time.

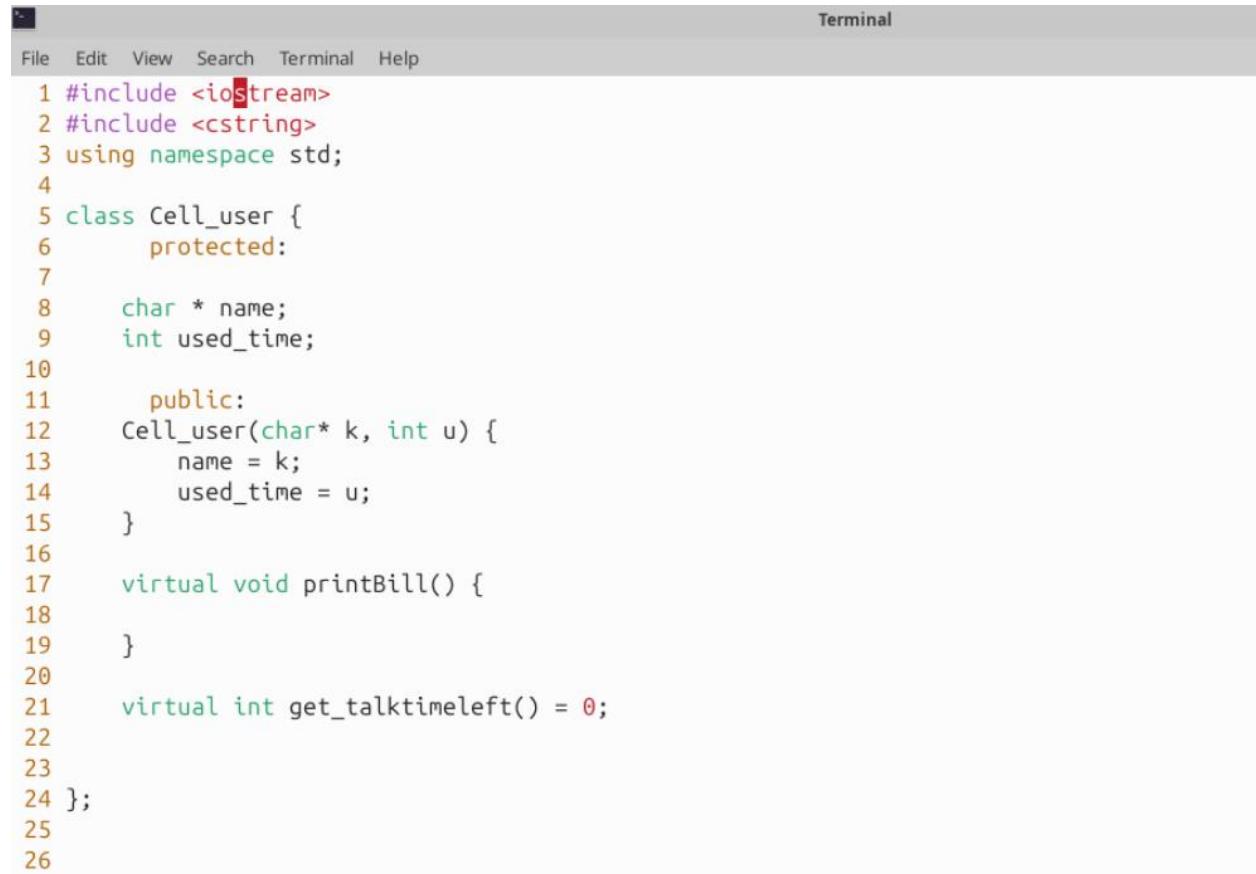
Define a class Cell\_user as a base class and derive the hierarchy of classes. Define member functions

and override them wherever necessary to

(I) retrieve the talk time left for each user.

(ii) print the bill in a proper format containing all the information for the post paid user.

CODE-



The image shows a terminal window with the title "Terminal". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The code is displayed in the terminal window:

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 class Cell_user {
6     protected:
7
8     char * name;
9     int used_time;
10
11     public:
12     Cell_user(char* k, int u) {
13         name = k;
14         used_time = u;
15     }
16
17     virtual void printBill() {
18
19     }
20
21     virtual int get_talktimeleft() = 0;
22
23
24 };
25
26
```

```

27 class Prepaid : public Cell_user {
28     private:
29     int total_time;
30
31     public:
32
33     Prepaid(char* k, int total, int used) : Cell_user(k, used) {
34         total_time = total;
35     }
36
37     int get_talktimeleft() override
38     {
39         return total_time - used_time;
40     }
41
42
43 };
44
45 class Postpaid : public Cell_user {
46     private:
47
48     int free_time;
49     float rate_per_pulse;
50
51     public:
52
53     Postpaid(char* k, int freetalk, int used) : Cell_user(k, used) {
54         free_time = freetalk;
55         rate_per_pulse = 1.90;
56     }
57
58     int get_talktimeleft() override {
59         if (used_time <= free_time)
60             return free_time - used_time;
61         else
62             return 0;
63     }
64
65     void printBill() override {
66         cout << "postpaid bill" << endl;
67         cout << "name - " << name << " " << "free ttime" << free_time << " mins" << endl;
68         cout << "used ttime" << used_time << " mins" << endl;
69
70         if (used_time > free_time)
71         {
72
73             int extra = used_time - free_time;
74
75             float bill = extra * rate_per_pulse;
76
77             cout << "extra pulse " << extra << " << "rate per puls" << rate_per_pulse << endl;
78
79             cout << "total bill==" << bill << endl;
80
81         } else {
82             cout << "total bill is 0" << endl;
83         }
84
85
86     }
87
88 };

```

```
90
91 int main() {
92     Prepaid p1("karan", 300, 120);
93     Postpaid p2("sush", 200, 260);
94
95     cout << endl "prepaid user" << endl;
96     cout << "talk time left = " << p1.get_talktimeleft() << " mins " << endl;
97
98     cout << endl << "postpaid user " << endl;
99
100    cout << "talk time left = " << p2.get_talktimeleft() << " min" << endl;
101    p2.printBill();
102
103
104 }
```

-- INSERT --