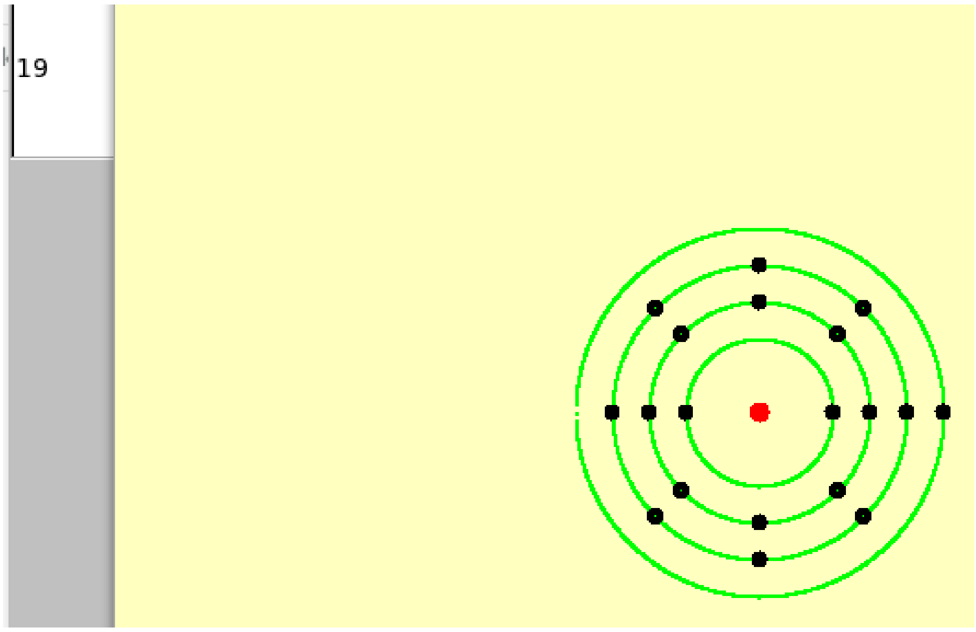
**Department of Computer Science and Engineering, IIT Bombay**
**CS 152 Abstractions and Paradigms for Programming**
**Quiz I Tuesday Feb. 11, 2020. 8:25-9:25 25 Marks Roll No:**

**Problem:** A code of a small application developed by a team in a basic programming laboratory is supplied in the attachment. A screenshot with sample input is given first.You are to thoroughly **analyze/criticise** this code from the point of view of an effective and neat application of abstractions and paradigm that we have touched upon in our discussions. Make a detailed report in the answerbook with **problems and their solutions**. Find as many problems as possible and provide your corrections for them. [Hints: classes, objects, inheritance, functions, polymorphism, superclass, subclass, abstract class, interface, types, signatures, indentation, parameters, variables, names, locals, globals, file data, data, control, loops, conditions ..]

**Instruction:** In your analysis, refer to code sections giving them tags/lables *(A,B,C..etc.)***.** **Mark relevant code sections, tag them clearly**, **and use those tags** in your analysis report. The question paper needs to be returned back along with the answer book since you are tagging parts of code and using those tags in your solutions.

**Input File and Screenshot of a sample run:** A cropped screenshot of a run of this application is given below. The left window is the input window, and the right one is the output produced. Also, sample contents of the three files used in this application are listed:



**CONTENTS OF INPUT FILES USED**

| **elements**<br>(names) | **elements2**<br>(ionization energy) | **elements3**<br>(atomic weight) | **elements4**<br>(discovery year) |
|---|---|---|---|
| Hydrogen | 13.5984 | 1.0079 | 1776 |
| Helium | 24.5874 | 4.0026 | 1895 |
| Lithium | 5.3917 | 6.941 | 1817 |
| Beryllium | 9.3227 | 9.0122 | 1797 |
| Boron | 8.298 | 10.811 | 1808 |
| Carbon | 11.2603 | 12.0107 | ancient |

| | | | |
|---|---|---|---|
| Nitrogen | ..... | ... | |

```
#include <FL/Fl.H> // header files
#include <FL/Fl_Window.H>
#include <FL/Fl_Box.H>
#include <iostream>
#include <string>
#include <fstream>
#include <math.h>
#include <cstdlib>
#include <ctime>
#include<FL/Fl_Int_Input.H>

using namespace std;
int m;
void f(Fl_Widget*w) {
Fl_Window *window = new Fl_Window (1000,700);
Fl_Int_Input *p=(Fl_Int_Input*)w;
const char *o=p->value();
m=atoi(o);
//m now has the value entered in the textbox

int n,y=20,q;
float x=-4,j,t;
string a[118],b[118],c[118],d[118];
ifstream e,f,g,h;

Fl_Box *b1;

f.open("elements");
for( int i=0;i<118;i++) // input names of elements from file "elements" into an array
f>>a[i];
f.close();

for( int i=0;i<118;i++) // input names of elements from file "elements" into an array
//cout<<a[i].c_str()<<endl;

e.open("elements3");
for( int i=0;i<118;i++) // input discovery year of elements from file "elements3" into an
array
e>>b[i];
e.close();

g.open("elements2");
for( int i=0;i<118;i++) // input atomic weight of elements from file "elements2" into an
array
g>>c[i];
g.close();
```

```
h.open("elements4");
for( int i=0;i<118;i++) //input ionisation energy of elements from file"elements4"into an
array
h>>d[i];
h.close();

srand(time(NULL));

q=(random()%4)+212;
window->color (q); // set its color

for(int n=1;n<=m;n++) {
if(n==1||n==3||n==11||n==19||n==37||n==55||n==87) {
if(n==1) j=7;
if(n==3) j=6.28318/(float)2;
if(n==11) j=6.28318/(float)8; //j= angle between adjacent electrons in same shell
if(n==19) j=6.28318/(float)8;
if(n==37) j=6.28318/(float)18;
if(n==55) j=6.28318/(float)18;
if(n==87) j=6.28318/(float)32;
float t=j;

while(j<=6.28319) {

for(int i=4;i>0;i-=0.05) {
for(x=-i;x<=i;x+=0.05) {
b1 = new Fl_Box (350+y*cos(j)+x,350+y*sin(j)+sqrt(i*i-x*x),1,1, "");
b1->box(FL_UP_BOX);
b1->box(FL_FLAT_BOX); //creates electron in the previous(penultimate) shell
b1->color (FL_BLACK);

b1 = new Fl_Box (350+y*cos(j)+x,350+y*sin(j)-sqrt(i*i-x*x),1,1, "");
b1->box(FL_UP_BOX);
b1->box(FL_FLAT_BOX);
b1->color (FL_BLACK);
}
}
j+=t;
}

y+=20;
for(x=-y;x<=y;x+=0.1) {

b1 = new Fl_Box (350-y,350,2,2, "");
b1->box(FL_UP_BOX);
b1->box(FL_FLAT_BOX);
b1->color (FL_GREEN);

b1 = new Fl_Box (350+y,350,2,2, "");
b1->box(FL_UP_BOX);
b1->box(FL_FLAT_BOX);
b1->color (FL_GREEN);

b1 = new Fl_Box (350+x,350+sqrt(y*y-x*x),2,2, ""); // creates upper half of shell
b1->box(FL_UP_BOX);
b1->box(FL_FLAT_BOX);
b1->color (FL_GREEN);
//creates lower half of shell
b1 = new Fl_Box (350+x,350-sqrt(y*y-x*x),2,2, "");
b1->box(FL_UP_BOX);
b1->box(FL_FLAT_BOX);
b1->color (FL_GREEN);
}
}
}
if(m>0) j=6.28318/(float)m;
if(m>2) j=6.28318/(float)(m-2);
if(m>10) j=6.28318/(float)(m-10);
if(m>18) j=6.28318/(float)(m-18); // m-i= no of electrons in valence shell
if(m>36) j=6.28318/(float)(m-36);
if(m>54) j=6.28318/(float)(m-54);
if(m>86) j=6.28318/(float)(m-86);
t=j;

while(j<=6.28319) {

for(int i=4;i>0;i-=0.05) {
for(x=-i;x<=i;x+=0.05) {
b1 = new Fl_Box (350+y*cos(j)+x,350+y*sin(j)+sqrt(i*i-x*x),1,1, "");
b1->box(FL_UP_BOX);
b1->box(FL_FLAT_BOX);
b1->color (FL_BLACK);
// It fills electrons in valence shell
b1 = new Fl_Box (350+y*cos(j)+x,350+y*sin(j)-sqrt(i*i-x*x),1,1, "");
b1->box(FL_UP_BOX);
b1->box(FL_FLAT_BOX);
b1->color (FL_BLACK);
}
}
j+=t;
}
for(y=5;y>0;y-=0.1) {
for(x=-y;x<=y;x+=0.1) {
b1 = new Fl_Box (350+x,350+sqrt(y*y-x*x),1,1, "");
b1->box(FL_UP_BOX);
b1->box(FL_FLAT_BOX);
b1->color (FL_RED); // It creates nucleus of atom
```

```
b1 = new Fl_Box (350+x,350-sqrt(y*y-x*x),1,1, "");
b1->box(FL_UP_BOX);
b1->box(FL_FLAT_BOX);
b1->color (FL_RED);
}
}
window->end();
window->show();
}
int main (int argc, char *argv[]) {

Fl_Window *window1 = new Fl_Window(500,500);
Fl_Int_Input *inp = new Fl_Int_Input(0,0,500,100,"Enter atomic number here");
inp->when(FL_WHEN_ENTER_KEY);
inp->callback(f);
window1->end();
window1->show();

return Fl::run();
}
```