

DATA STRUCTURES LAB

Ex. No. 9	IMPLEMENTATION OF HASHING
06-09-2017	

Question:

To implement the working of implementation of hashing.

Algorithm:

1. Start.
2. For inserting an element,
3. If it is the first element, it is already sorted. return 1;
4. Pick next element.
5. Compare with all elements in the sorted sub-list
6. Shift all the elements in the sorted sub-list that is greater than the value to be sorted
7. Insert the value
8. Repeat until list is sorted
9. For selection sort, follow the steps:
10. Set MIN to location 0
11. Search the minimum element in the list
12. Swap with value at location MIN
13. Increment MIN to point to next element
14. Repeat until list is sorted
15. End.

DATA STRUCTURES LAB

Program:

```
#include<iostream>

#include<cstdlib>

using namespace std;

int a[5]={-1,-1,-1,-1,-1};

void insert(int);

void display();

int delet(int);

int main(){

int ch;

cout<<"Operations are:\n1.Insert\n2.Display\n3.Delete\n4.Exit\n";

while(1){

cout<<"\n*****\n";

cout<<"Enter your choice: ";

cin>>ch;

cout<<"\n*****\n";

switch(ch){

case(1):

int x;

cout<<"Enter the data to be inserted: ";

cin>>x;

insert(x);
```

DATA STRUCTURES LAB

```
    break;
case(2):
    display();
    break;
case(3):
    int y;
    cout<<"Enter the data to be deleted: ";
    cin>>y;
    delet(y);
    break;
case(4):
    exit(0);
    break;
default:
    cout<<"Enter valid option!"<<endl;
    break;
}
}
}

void insert(int x){
    int pos=x%5;
    if(a[pos]==-1){
```

DATA STRUCTURES LAB

```
a[pos]=x;
cout<<"Data has been inserted!"<<endl;
}
else{
    int i=((pos+1)%5);
    for(i;i!=pos;i=((i+1)%5)){
        if(a[i]==-1){
            a[i]=x;
            cout<<"Data has been inserted!"<<endl;
            break;
        }
    }
    if(i==pos){
        cout<<"Hash is Full!"<<endl;
    }
}
}

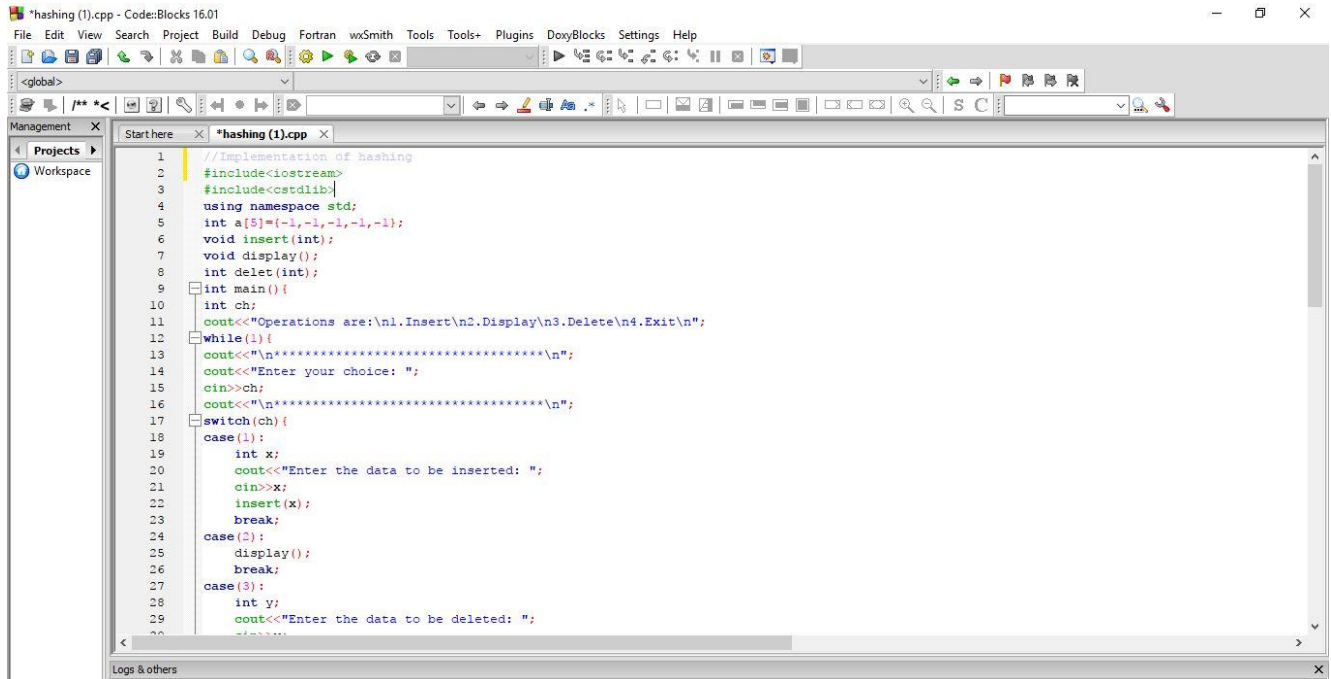
void display(){
    for(int i=0;i<5;i++){
        cout<<a[i]<<" ";
    }
}
```

DATA STRUCTURES LAB

```
int delet(int y){
int pos=y%5;
if(a[pos]==y){
    cout<<"Data was at position: "<<pos<<endl;
    cout<<"Data has been deleted!"<<endl;
    a[pos]=-1;
}
else{
    int z=0;
    for(int i=(pos+1)%5;i!=pos;i=(i+1)%5){
        if(a[i]==y){
            cout<<"Data was at position: "<<i<<endl;
            cout<<"Data has been deleted!"<<endl;
            a[i]=-1;
        }
        else{
            z=-1;
        }
    }
    if(z==-1){
        cout<<"Data was not found!"<<endl;
    }
}
```

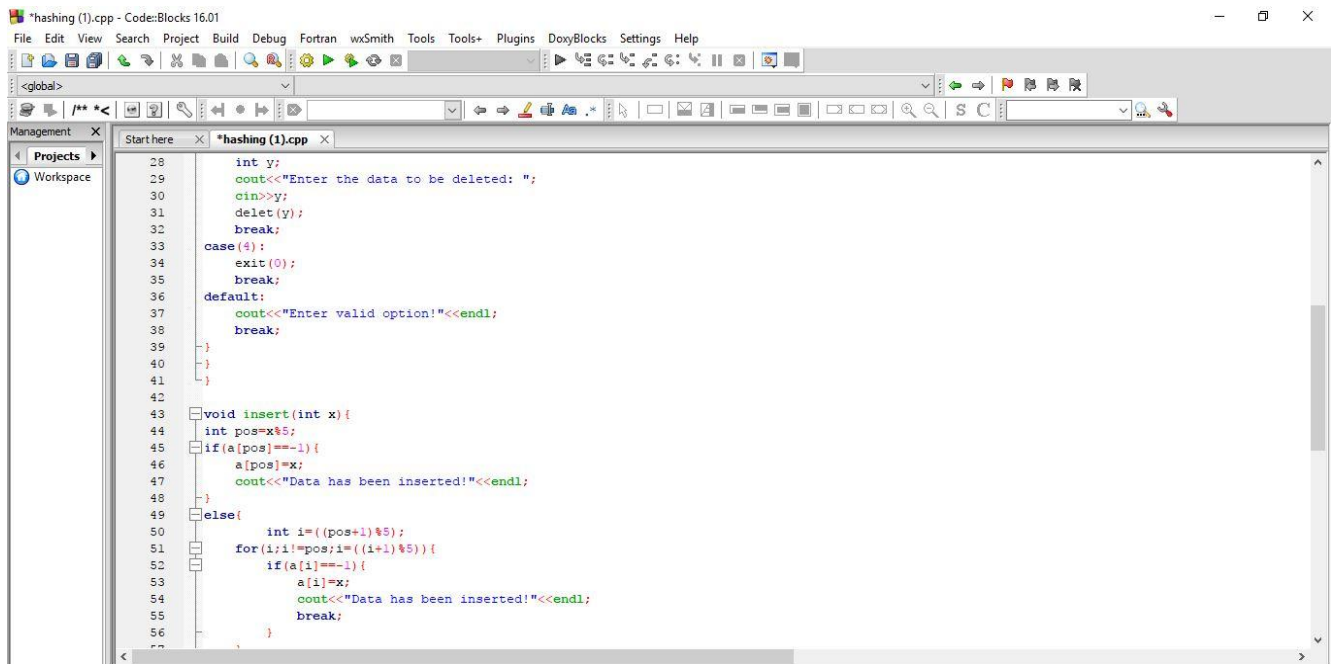
DATA STRUCTURES LAB

Output:



The screenshot shows a C++ IDE with the file `*hashing (1).cpp` open. The code defines a hash table with 5 slots, each containing an array of 5 elements. The `main` function prompts the user to choose an operation (1: Insert, 2: Display, 3: Delete, 4: Exit). The `switch` statement handles these operations. The `insert` function is partially visible at the bottom of the screen.

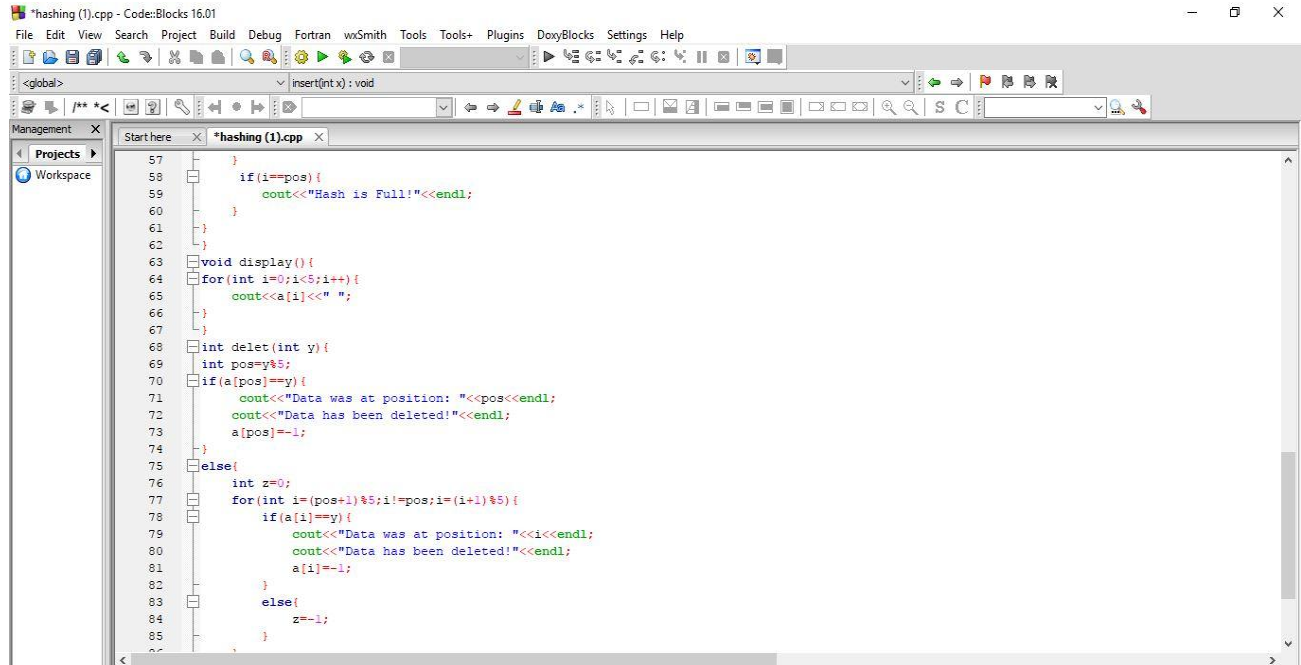
```
1 //Implementation of hashing
2 #include<iostream>
3 #include<cstdlib>
4 using namespace std;
5 int a[5][5]={-1,-1,-1,-1,-1};
6 void insert(int);
7 void display();
8 int delet(int);
9 int main(){
10     int ch;
11     cout<<"Operations are:\n1.Insert\n2.Display\n3.Delete\n4.Exit\n";
12     while(1){
13         cout<<"\n*****\n";
14         cout<<"Enter your choice: ";
15         cin>>ch;
16         cout<<"\n*****\n";
17         switch(ch){
18             case(1):
19                 int x;
20                 cout<<"Enter the data to be inserted: ";
21                 cin>>x;
22                 insert(x);
23                 break;
24             case(2):
25                 display();
26                 break;
27             case(3):
28                 int y;
29                 cout<<"Enter the data to be deleted: ";
30                 cin>>y;
```



The screenshot shows the continuation of the C++ code from the previous image. It includes the `delet` function, the `default` case of the `switch` statement, and the `insert` function. The `insert` function uses a separate array `a` to store the data and a `pos` variable to track the current position in the hash table. It checks for collisions and inserts the data at the next available slot.

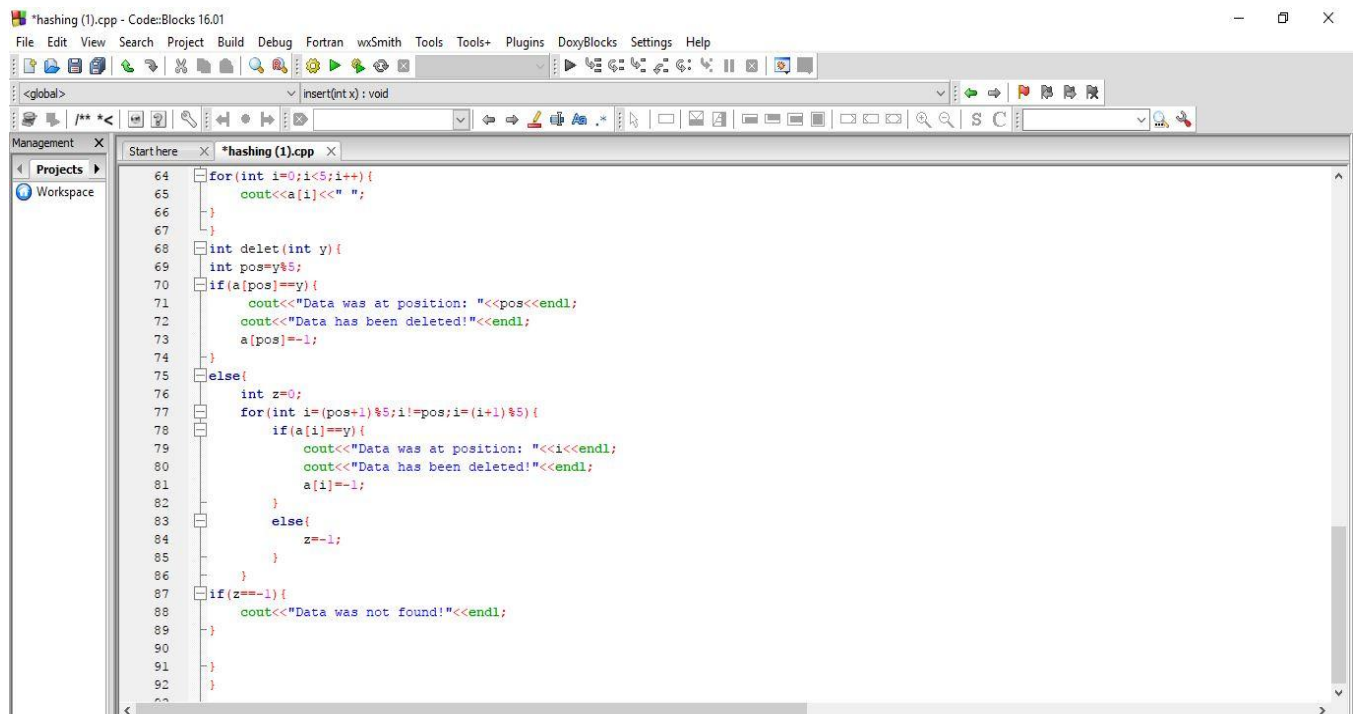
```
28     int y;
29     cout<<"Enter the data to be deleted: ";
30     cin>>y;
31     delet(y);
32     break;
33     case(4):
34         exit(0);
35         break;
36     default:
37         cout<<"Enter valid option!"<<endl;
38         break;
39 }
40 }
41 }
42
43 void insert(int x){
44     int pos=pos;
45     if(a[pos]==-1){
46         a[pos]=x;
47         cout<<"Data has been inserted!"<<endl;
48     }
49     else{
50         int i=((pos+1)%5);
51         for(i;i!=pos;i=((i+1)%5)){
52             if(a[i]==-1){
53                 a[i]=x;
54                 cout<<"Data has been inserted!"<<endl;
55                 break;
56             }
57         }
58     }
59 }
```

DATA STRUCTURES LAB



The screenshot shows the Code::Blocks IDE with a C++ file named *hashing (1).cpp. The code implements a hash table with a display function. The hash table is an array of 5 slots. The display function iterates through the slots and prints the contents. The code is as follows:

```
57 }
58 if(i==pos){
59     cout<<"Hash is Full!"<<endl;
60 }
61 }
62
63 void display(){
64     for(int i=0;i<5;i++){
65         cout<<a[i]<<" ";
66     }
67 }
68 int delet(int y){
69     int pos=y%5;
70     if(a[pos]==y){
71         cout<<"Data was at position: "<<pos<<endl;
72         cout<<"Data has been deleted!"<<endl;
73         a[pos]=-1;
74     }
75     else{
76         int z=0;
77         for(int i=(pos+1)%5;i!=pos;i=(i+1)%5){
78             if(a[i]==y){
79                 cout<<"Data was at position: "<<i<<endl;
80                 cout<<"Data has been deleted!"<<endl;
81                 a[i]=-1;
82             }
83             else{
84                 z=-1;
85             }
86         }
87     }
88 }
```



The screenshot shows the Code::Blocks IDE with a C++ file named *hashing (1).cpp. The code implements a hash table with a delete function. The hash table is an array of 5 slots. The delete function takes a value y and checks if it is in the hash table. If it is, it prints the position and deletes the data. If it is not, it prints "Data was not found!". The code is as follows:

```
64 for(int i=0;i<5;i++){
65     cout<<a[i]<<" ";
66 }
67 }
68 int delet(int y){
69     int pos=y%5;
70     if(a[pos]==y){
71         cout<<"Data was at position: "<<pos<<endl;
72         cout<<"Data has been deleted!"<<endl;
73         a[pos]=-1;
74     }
75     else{
76         int z=0;
77         for(int i=(pos+1)%5;i!=pos;i=(i+1)%5){
78             if(a[i]==y){
79                 cout<<"Data was at position: "<<i<<endl;
80                 cout<<"Data has been deleted!"<<endl;
81                 a[i]=-1;
82             }
83             else{
84                 z=-1;
85             }
86         }
87     }
88     if(z==-1){
89         cout<<"Data was not found!"<<endl;
90     }
91 }
92 }
```

DATA STRUCTURES LAB

"C:\Users\JUSTIN PAUL\Downloads\hashing (1).exe"

Operations are:

- 1.Insert
- 2.Display
- 3.Delete
- 4.Exit

Enter your choice: 1

Enter the data to be inserted: 12
Data has been inserted!

Enter your choice: 1

Enter the data to be inserted: 10
Data has been inserted!

Enter your choice: 1

Enter the data to be inserted: 23
Data has been inserted!

Enter your choice: 1

Enter the data to be inserted: 34
Data has been inserted!

Enter your choice: 1

Enter the data to be inserted: 45
Data has been inserted!

Enter your choice: 1

"C:\Users\JUSTIN PAUL\Downloads\hashing (1).exe"

Enter the data to be inserted: 56
Hash is Full!

Enter your choice: 2

10 45 12 23 34
Enter your choice: 3

Enter the data to be deleted: 10
Data was at position: 0
Data has been deleted!

Enter your choice: 2

-1 45 12 23 34
Enter your choice: 3

Enter the data to be deleted: 23
Data was at position: 3
Data has been deleted!

Enter your choice: 2

-1 45 12 -1 34
Enter your choice: 4

Process returned 0 (0x0) execution time : 77.177 s

Press any key to continue.

VIDEO URL:

<https://youtu.be/TcTiq5PtQbs>

RESULT:

The program of implementation of hash function is implemented successfully and the output is verified.