Experiment 2

AIM: Write a program which defines a class Employee with attributes employee number, and employee name. Create a list of employees and sort them on the basis on name and id.

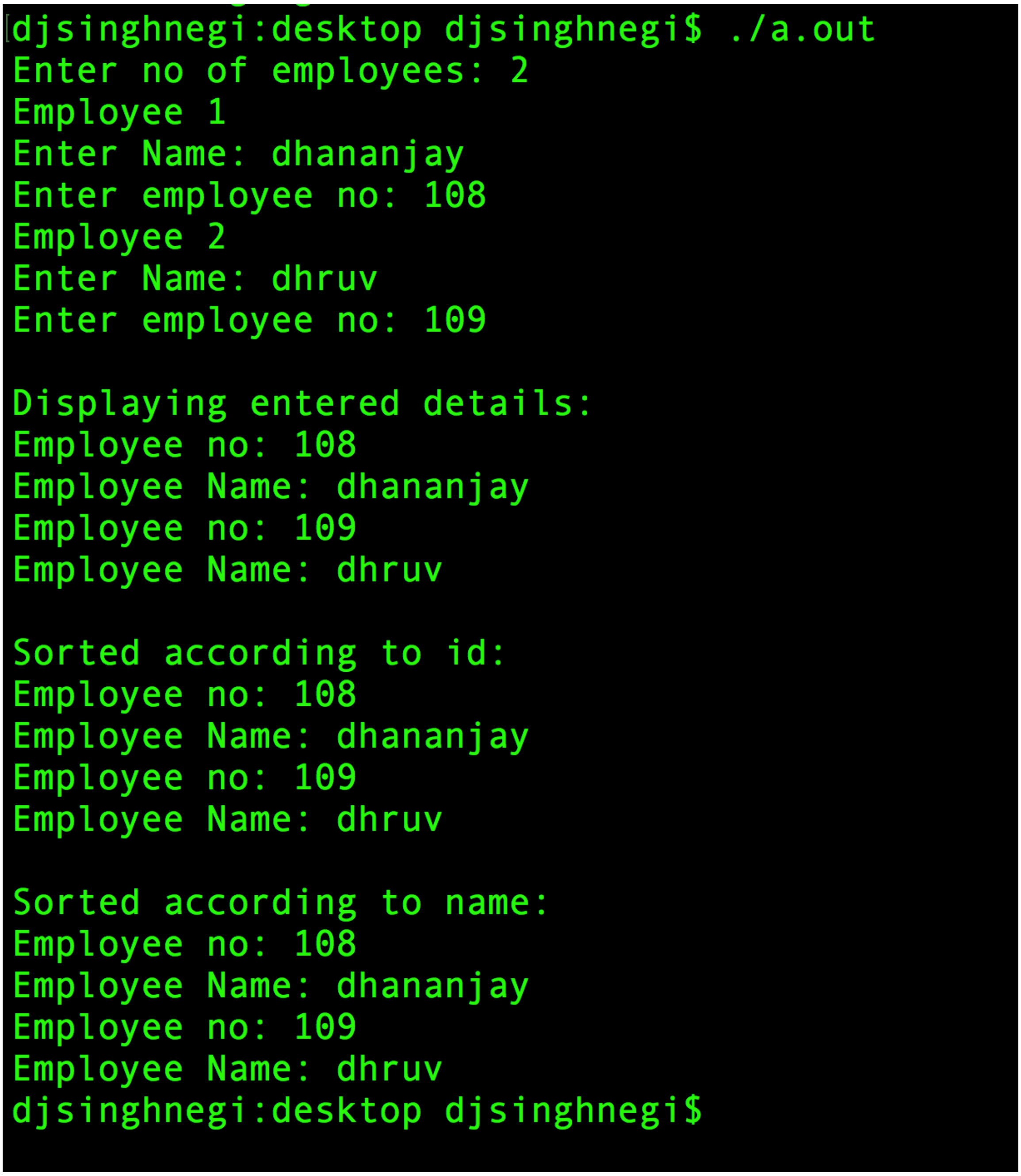
Theory:

Getters and setters are used to get and set private variables. The sort function sorts an array according to the comparator provided. By default it sorts the array in ascending order.

Code:

```
#include < iostream >
#include < string >
#include < algorithm >
using namespace std;
class Employee {
    int employeeNo;
    string employeeName;
    public: void getData() {
        cout << "Enter Name: ";
        cin.ignore();
        getline(cin, employeeName);
        cout << "Enter employee no: ";
        cin >> employeeNo;
    }
    void display() {
        cout << "Employee no: " << employeeNo << endl;
        cout << "Employee Name: " << employeeName << endl;
    }
    int getID() {
        return employeeNo;
    }
    string getName() {
        return employeeName;
    }
};
bool cmp_id(Employee a, Employee b) {
    return a.getID() < b.getID();
}
bool cmp_name(Employee a, Employee b) {
    return a.getName() < b.getName();
}
int main() {
    int n;
    cout << "Enter no of employees: ";
    cin >> n;
    Employee emps[n];
    for (int i = 0; i < n; i++) {
        cout << "Employee " << i + 1 << endl;
        emps[i].getData();
    }
    cout << "\nDisplaying entered details:\n";
    for (int i = 0; i < n; i++) emps[i].display();
    cout << "\nSorted according to id:\n";
    sort(emps, emps + n, cmp_id);
    for (int i = 0; i < n; i++) emps[i].display();
    cout << "\nSorted according to name:\n";
    sort(emps, emps + n, cmp_name);
    for (int i = 0; i < n; i++) emps[i].display();
    return 0;
}
```

Output:



```
djsinghnegi:desktop djsinghnegi$ ./a.out
Enter no of employees: 2
Employee 1
Enter Name: dhananjay
Enter employee no: 108
Employee 2
Enter Name: dhruv
Enter employee no: 109

Displaying entered details:
Employee no: 108
Employee Name: dhananjay
Employee no: 109
Employee Name: dhruv

Sorted according to id:
Employee no: 108
Employee Name: dhananjay
Employee no: 109
Employee Name: dhruv

Sorted according to name:
Employee no: 108
Employee Name: dhananjay
Employee no: 109
Employee Name: dhruv
djsinghnegi:desktop djsinghnegi$
```

Discussion:

The class Employee stores details like, employee number, and employee name. An array of Employee objects, named, emps is created. The user then inputs the employee details. The program then displays the entered details and also shows a sorted list of employees based on employee number and name.