C++ Lab Record

Program 1:

Write a C++ Program to display Names, Roll No., and grades of 3 students who have appeared in the examination. Declare the class of name, Roll No. and grade. Create an array of class objects. Read and display the contents of the array.

```
  # include <iostream>
  #include <string>
  using namespace std;
  class Student{
          public:
                  string name;
                  int rollno;
                  int marks;
                  void read();
                  void display();
          }s[3];
void Student :: display(){
                  for (int i = 0; i < 3; i++){
                          cout << s[i].name << endl;
                          cout << s[i].rollno <<endl;
                          cout << s[i].marks<< endl;
          }
}
int main(){
          s[3].read();
          s[3].display();
          cout << s[0].name;
          return 0;
}
```

Output:

```
Madeeha
1247
60
Rida
1210
70
divya
1248
80
Anu
1242
60
Saleha
1245
70
Meena
1226
80
```

Program 2:

Given that an EMPLOYEE class contains following members: data members: Employee number, Employee name, Basic, DA, IT, Net Salary and print data members.

```
#include <iostream>
#include <string>
using namespace std;
class Employee{
        public:
                int e_no;
                string e_name;
                int  basic();
                int  da;
                int  it;
                int  net_sal;
                void read(){
                        cout << “Enter Employee name:”;
                        cin >>e_no;
                        cout << “Enter the Employee name:”;
                        cin >> e_name;
                        cout << “Enter basic salary:”;
                        cin >> basic;
                        cout >> “Enter DA:”;
                        cin >> da;
                        cout << “Enter IT:”;
                        cin >> it;
                }
                void  cal_net_sal(){
                                Net_sal = basic + da – it;
                }
                void display(){
                                cout << “Employee number:” << e_no <<endl;
                                cout << “Emplolyee name:” << e_name<<endl;
                                cout << “Basic salary:” << basic << endl;
                                cout << “DA:” << da << endl;
                                cout << “IT:” << it << endl;
                                cout << “Net salary:” << net_sal<< endl;
                }
}
int  main(){
        Employee e;
        e.read();
        e.cal_net_sal();
        e.display();
        return 0;
}
```

Output:

```
Enter Employee number: 1
Enter Employee name: Madeeha
Enter Basic salary: 10000
Enter DA: 2000
Enter IT: 1000
Employee number : 1
Employee name : 10000
DA: 2000
IT: 1000
Net Salary : 11000
```

Program 3:

Write a C++ program to read the data of N employee and compute Net salary of each employee (DA=52% of Basic and Income Tax (IT) =30% of the gross salary).

```
#include <iostream<
#include <string>
using namespace std;
Class Employee{
        Public:
                int e_id;
                string name;
                float bs;
                float da;
                float it;
                float net_sal;
                float gross;
                void read(){
                        cout << “Enter Employee id, name, basic, salary:”;
                        cin >> e.id >> name >> bs;
                }
                void cal_da(){
                        da = 0.52 * bs;
                }
                void cal_gross(){
                        gross = bs + da;
                }
                void cal_it(){
                        it = 0.30 * gross;
                }
                void cal_net_sal(){
                net_sal = gross-it;
                }
                void display(){
                      cout << “Employee id:” << e.id << endl;
                      cout << “Employee name: “ << name << endl;
                      cout << “DA: “ << da << endl;
                      cout << “IT: “ << it << endl;
                      cout << “Gross Salary:” << gross << endl;
                      cout << “Net salary: “ << net_sal << endl;
               }
};
int main(){
Employee e[10];
int i;
for(i = 0; i < 10; i++){
        e[i].read();
        e[i].cal_da();
        e[i].cal_gross();
        e[i].cal_it();
        e[i].display();
        return 0;
        }
}
```

Output

```
Enter Employee id, name, basic, salary:
Madeeha
10000
Employee id: 1
Employee name: seema
Basic salary: 10000
DA: 5200
IT: 4560
Gross Salary: 152000
Net Salary: 10640
```

Program 4:

Write a C++ program to declare Struct. Initialize and display contents of member Variables.

```
#include <iostream>
#include <string>
using namespace std;
struct Student{
        string name;
        int rollno;
        int marks;
}s1;
void display(){
        cout << s1.name << endl;
        cout << s1.rollno << endl;
        cout << s1.marks << endl;
}
int main(){
         cout << “Enter name: “;
         cin >> s1.name;
         cout << “Enter rollno:”;
         cin >> s1.rollno;
         cout << “Enter marks:”;
         cin >> s1.marks;
         display();
}
```

Output

```
Enter name: Madeeha
Enter rollno: 1247
Enter marks: 80
Madeeha
1247
80
```

Program 5:

Write a C++ program to declare a class. Declare pointer to class. Initialize and display the contents of the class member.

```
 #include <iostream>
 #include <string>
 using namespace std;
 class  student{
        int rollno;
        string  name;
        string  Branch;
        string grade;
        public:
        void read(){
                cout << “Enter rollno, name, Branch, grade:” << endl;
                cin >> rollno >> name >> Branch >> grade;
        }
        void display(){
                  cout << “rollno: “ << rollno << endl;
                  cout << “name:” ,, name << endl;
                  cout << “Branch:” << branch<< endl;
                  cout << “grade:” << grade << endl;
        }
};
int  main(){
        student s1;
        student *ptr;
        ptr = &s1;
        ptr -> read();
        ptr -> display();
        return 0;
}
```

Output

```
Enter rollno, name, Branch, grade:
1247
Madeeha
IT
A
rollno: 1247
name: Madeeha
Branch: IT
grade: A
```

Program 6:

Write a C++ program to use scope resolution operator. Display the various values of the same variables declared at different scope levels.

```
#include <iostream>
#include <string>
using namespace std;
class A1{
        public:
        int a;
        A1(){
                a1 = 10;
        }
        void display(){
                cout << a1 << endl;
        }
};
        class A2{
        public:
        int a2;
        A2(){
                a2 = 20;
        }
        void display(){
                cout << a2 << endl;
        }
};
class A3: public A1, public A2{
public:
        int a3;
        A3(){
        a3 = 30;
        }
        void max();
        void display3(){
                cout << a3 << endl;
                A1 :: display();
                A2 :: display();
        }
};
void A3 :: max(){
        if ((a1 > a2) && (a1 > a3)){
                cout << a1 << endl;
}
        else if ((a2 > a1) && (a2 > a3)){
                cout << a2 << endl;
        }
}
int i = 100;
int main(){
        int i = 200;
        A3 obj;
        obj.display3();
        obj.max();
        cout << i << endl;
        return 0;
}
```

```
Output:
30
10
20
30
200
100
```

Program 7:

Write a C++ program using this pointer.

```
#include <iostream>
using namespace std;
class Test{
	private:
		int x;
	public:
		void setX(int x){
			this -> x = x;
		}
		void print(){
			cout << “x = “ << x << endl;
		}
};
int main(){
	Test obj;
	int x = 20;
	obj.setX(x);
	obj.print();
	return;
}
```

Output:

```
x = 20
```

Program 8 :

Write a C++ program on **Friend** Class **&**Friend Function

```
#include <iostream>
using  namespace  std;
class A {
private:
    int a;

public:
    A() { a = 0; }
    friend class B;
    void  showB(B&);
};
class B {
private:
    int b;

public:
    B(){ int b = 0;}
    void showA(A& x)
    {

        cout << "A::a=" << x.a;
    }
    friend void A::showB(B& x);
};
void A::showB(B& x)
{

       cout << "B::b = " << x.b;
}


int main()
{
    A a;
    B b;
     b.showA(a);
    a.showB(b);
    return 0;
}
```

Output

```
A::a=0
B::b = 0
```

Program 9:

Program to demonstrate Constructors & Destructors

```
  #include <iostream>
  using namespace std;
  class Line{
          public:
          void setLength(double len);
          double getLength(void);
          Line();
          ~Line();
          private:
                    double length;
  };
  Line:: Line(void){
          cout << “Object is being created” << endl;
  }
  Line :: ~Line(void){
          cout << “Object is being deleted: << endl;
  }
  void Lined :: setLength(double len){
          length = len;
}
double Line :: getLength(void){
          return length;
}
int main(){
          Line line;
          line.setLength(6.0);
          cout << “Length of line: “ << line.getLength() << endl;
          return 0;
}
```

Output

```
Object is being created
Length of line : 6
Object is being deleted
```

Program 10:

Write a C++ program to allocate memory using new operator

```
#include <iostream>
using namespace std;
int main(){
        int *ptr;
        ptr = new int;
        cout << “Number of bytes allocated to ptr is” << sizeof(ptr) << endl;
        *ptr = 100;
        cout << “Value at ptr is” << ptr << endl;
        return 0;
}
```

Output

```
Number of bytes allocated to ptr is 4
value at ptr is 1
```