Topic Date

INT105

Name: Rohan Kumar Sahu
Roll No: RM2041A01
Reg No: 12011878

SET-A

Q1. Create 5 objects of class Product, input its product code, name and quantity and write into the FILE. Later, display ~~de~~ the data of all objects by reading from the FILE.

Solution =
```
#include <iostream>
#include <fstream>
using namespace std;

class Product
{
public:
    string name, code;
    long int quantity;
    char c;
    void get()
    {
      fflush(stdin);
      cout << "Enter product name: ";
```

(3)

Topic Date

```
        getline (cin, name);
        cout << "Enter product code:";
        getline (cin, code);
        cout << "Enter quantity: ";
        cin>> Quantity;
        c='\n';
    }
    void display()
    {
        cout <<"\n Product name: " << name << endl;
        cout << "Product code: " << code << endl;
        cout << "Product quantity:" << quantity << endl;
    }
};

int main()
{
  int n;
  cout << "Enter number of products: ";
  cin>> n;
  Product * ptr = new Product [n];
  Product e;
  fstream file;
  file. open(" productDetails.txt", ios:: out);
  for (int i=0 ; i<n; i++)
  {
```

CRYSTAL

Teacher's Signature

```
        cout << "Product " << (i+1) << endl ;
        ptr[i]. get()
        file. write ((char*)& ptr[i], sizeof (ptr[i])) ;
    }
    file. close ;
    file. open ("productDetails. txt", ios:: in);
    file. seekg(0) ;
    file. read ((char*) & e , sizeof (e)) ;
    cout << "\n Displaying details" << endl ;
    while (file. eof() ==0)
    {
        e. display();
        file. read ((char*) &e , sizeof (e)) ;
    }
    file. close() ;
    return 0 ;
}
```

Q2. Choose data members for class Car to define its properties. Create 2 objects of class Cars and overload the operator 'greater than' > to compare both the objects concerning their data members.

Solution =
```
# include <iostream>
using namespace std;

class Car
```

Topic Date

```
{
public :
   float horse_power, mileage;
   Car()
   {
      cout << "Enter mileage of the car : ";
      cin >> mileage;
      cout << "Enter horse power of the car : ";
      cin >> horse_power;
   }
   Car operator > (Car c)
   {
      if (mileage > c. mileage)
      {
         cout << "Mileage of car 1 is more" << endl;
      }
      else if ( mileage == c. mileage)
      {
         cout << "Both cars have equal mileage" << endl;
      }
      else
      {
         cout << "Mileage of car 2 is more" << endl;
      }
      if (horse_power > c. horse_power)
      {
```

Topic Date

```
        cout << "Horse power of car 1 is more" << endl;
      }
      else if (horse_power == c.horse_power)
      {
         cout << "Both cars have same horse power" << endl;
      }
      else
      {
         cout << "Horse power of car 2 is more" << endl;
       }
    }
};

int main()
{
   cout << "car 1" << endl;
   car c1;
   cout << "car 2" << endl;
   car c2;
   cout << "\n";
   c1>c2;
   return 0;
}
```

Q3. What is the difference between Virtual Function and Pure Virtual Function?

CRYSTAL

Teacher's Signature

Is it always mandatory to implement or define all the pure virtual function of the base class into derived class? Justify your answer.

Solution =

Virtual function	Pure virtual function
• A virtual function is a member function of base class which can be redefined by derived class.	A pure virtual function is a member function of base class whose only declaration is provided in base class and should be defined class.
• Classes having virtual functions are not abstract.	Base class containing pure virtual function becomes abstract.
• Syntax: virtual <func_type> <func_name> () { // code }	Syntax: virtual <func_type> <func_name> () = 0;
• Definition is given in base class.	No definition is given in base class.
• Base class having virtual function can be instantiated i.e. its object can be made.	Base class having pure virtual function becomes abstract i.e. it cannot be instantiated.
• If derived class do not redefine virtual function of base class, then it does not affect compilation.	If derived class do not redefine pure virtual function of base class, then no compilation error but derived class also becomes ~~ast~~ abstract just like the base class.
• All derived class may or may not redefine virtual function of base class	All derived class must redefine pure virtual function of base class otherwise derived class also becomes abstract like base

Topic Date

```
#include <iostream>
using namespace std;

class Base
{
protected:
    string name, rollno;

public:
    virtual void get() = 0;
};

class Derived 1: public Base
{
public:
   void get()
   {
      cout << "Enter name: ";
      getline (cin, name);
      cout << "Enter roll no: ";
      getline ( cin ,rollno);
   }
   void display ()
   {
      cout << "\nThe name is" << name << endl;
      cout << " The roll no is " << rollno << endl;
```

Topic Date

```
    }
};

class Derived 2 : public Base
{
public :
        void display ()
        {
            cout << "The name is " << name << endl;
            cout << " The roll no is " << rollno << endl;
        }
};

int main()
{
    Derived1 ob1;
    ob1. get();
    ob1. display();
    Derived 2 ob2;   // The below lines will show an error, we need to redefine get function
    ob2. get() ;
    ob2. display();
    return 0;
}
```

Q4. Write a programme to demonstrate the Dynamic Memory Allocation with some

Topic Date

real-time example. Also, mention the possible ways of initializing and deallocating memory from dynamically created objects of the class.

Solution =

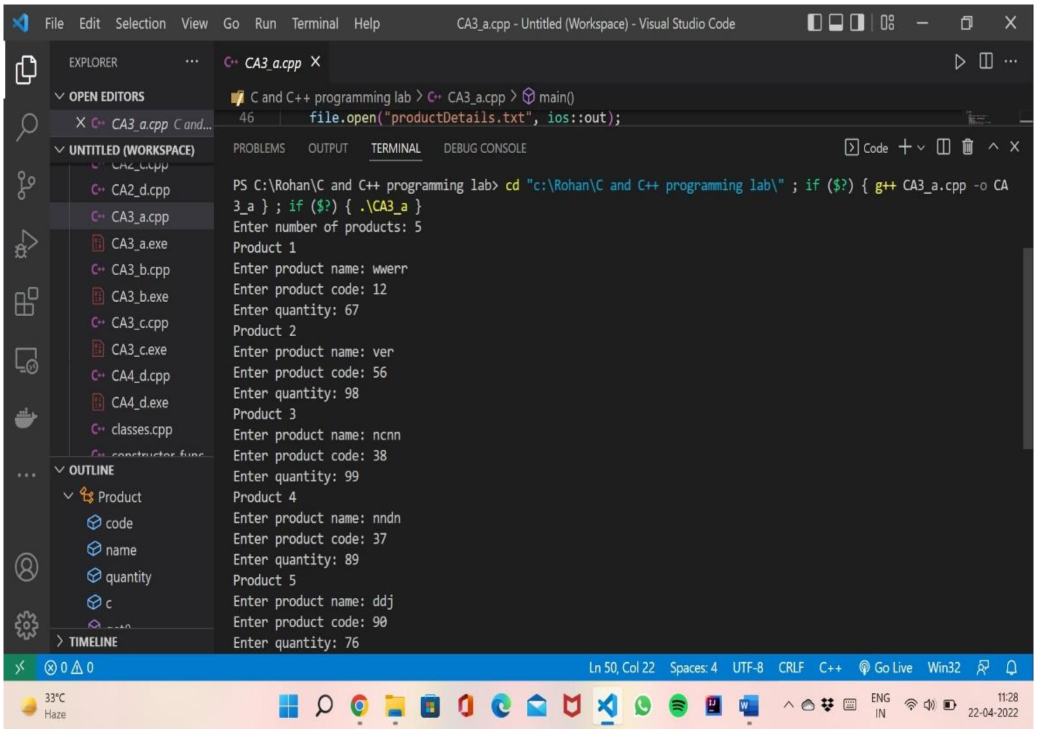
```
# include <iostream>
using namespace std;

class Employee
{
   string name;
   long long int contact;

public:
    void get()
    {
       fflush(stdin);
       cout << "Enter name: ";
       getline(cin, name);
       cout << " Enter contact number: ";
       cin >> contact;
    }
    void display ()
    {
        cout << "\n The name is " << name << endl;
        cout << " The contact number is " << contact << endl;
    }
};
```

Topic Date

```
int main()
{
  int n;
  cout << "Enter number of employees: ";
  cin >> n;
   Employee *p = new Employee [n];
   for (int i =0 ; i<n ; i++)
    {
      cout << "\n Employee " << (i+1) << endl;
      p[i]. get();
    }
   cout << "\n Displaying details " << endl;
   for (int i =0 ; i<n ; i++)
   {
      p[i]. display();
   }
   delete [] p;
   return 0;
}
```

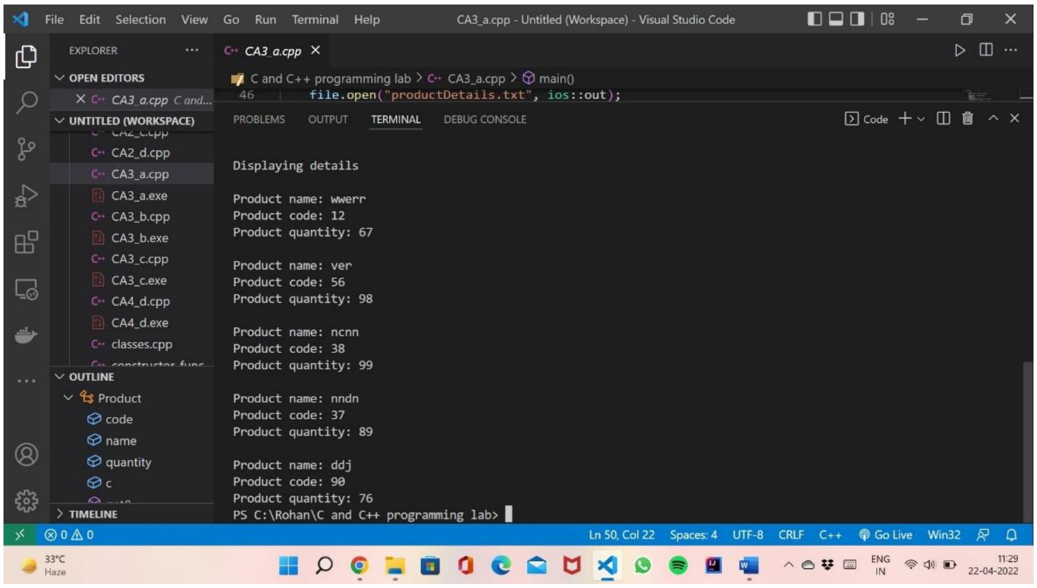
```
/*
INT105
CA3
Name: Rohan Kumar Saini
Roll No: RM2041A01
Reg No: 12011878
Question1
*/
#include <iostream>
#include <fstream>
using namespace std;

class Product
{
public:
    string name, code;
    long int quantity;
    char c;
    void get()
    {
        fflush(stdin);
        cout << "Enter product name: ";
        getline(cin, name);
        cout << "Enter product code: ";
        getline(cin, code);
        cout << "Enter quantity: ";
        cin >> quantity;
        c = '\n';
    }
    void display()
    {
        cout << "\nProduct name: " << name << endl;
        cout << "Product code: " << code << endl;
        cout << "Product quantity: " << quantity << endl;
    }
};

int main()
{
    int n;
    cout << "Enter number of products: ";
    cin >> n;
    Product *ptr = new Product[n];
    Product e;
    fstream file;
    file.open("productDetails.txt", ios::out);
    for (int i = 0; i < n; i++)
    {
        cout << "Product " << (i + 1) << endl;
```

```cpp
        ptr[i].get();
        file.write((char *)&ptr[i], sizeof(ptr[i]));
    }
    file.close();
    file.open("productDetails.txt", ios::in);
    file.seekg(0);
    file.read((char *)&e, sizeof(e));
    cout << "\nDisplaying details" << endl;
    while (file.eof() == 0)
    {
        e.display();
        file.read((char *)&e, sizeof(e));
    }
    file.close();
    return 0;
}
```



```cpp
        file.open("productDetails.txt", ios::out);
```

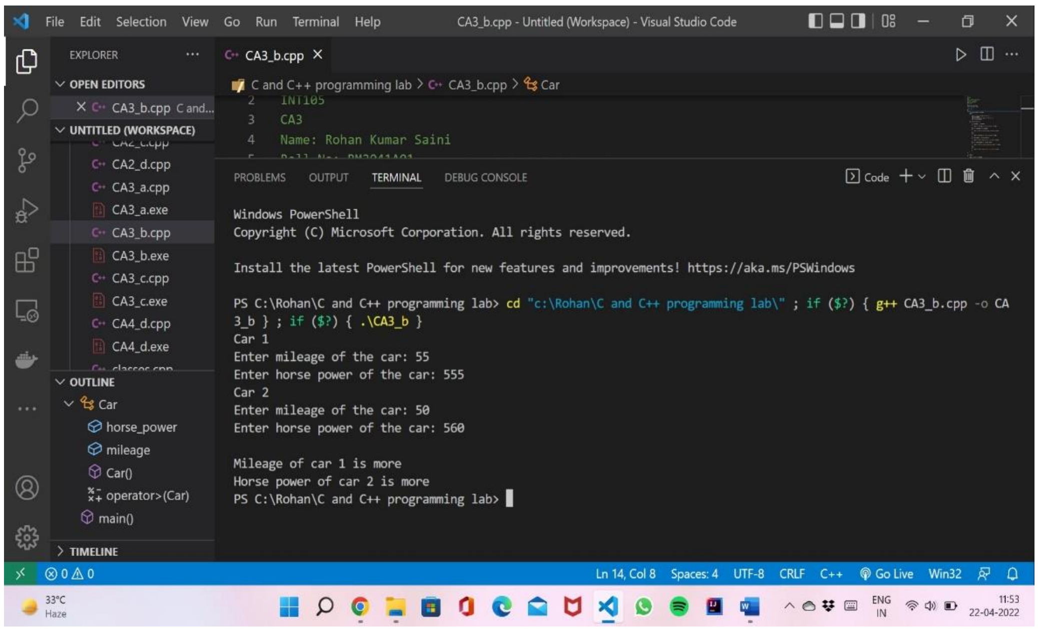
```
PS C:\Rohan\C and C++ programming lab> cd "c:\Rohan\C and C++ programming lab\" ; if ($?) { g++ CA3_a.cpp -o CA
3_a } ; if ($?) { .\CA3_a }
Enter number of products: 5
Product 1
Enter product name: wwerr
Enter product code: 12
Enter quantity: 67
Product 2
Enter product name: ver
Enter product code: 56
Enter quantity: 98
Product 3
Enter product name: ncnn
Enter product code: 38
Enter quantity: 99
Product 4
Enter product name: nndn
Enter product code: 37
Enter quantity: 89
Product 5
Enter product name: ddj
Enter product code: 90
Enter quantity: 76
```


File Edit Selection View Go Run Terminal Help
CA3_a.cpp - Untitled (Workspace) - Visual Studio Code
EXPLORER
OPEN EDITORS
CA3_a.cpp C and...
UNTITLED (WORKSPACE)
CA2_d.cpp
CA3_a.cpp
CA3_a.exe
CA3_b.cpp
CA3_b.exe
CA3_c.cpp
CA3_c.exe
CA4_d.cpp
CA4_d.exe
classes.cpp
OUTLINE
Product
code
name
quantity
c
TIMELINE
C and C++ programming lab > CA3_a.cpp > main()
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Code
Ln 50, Col 22 Spaces: 4 UTF-8 CRLF C++ Go Live Win32
33°C Haze
ENG IN
11:29
22-04-2022

```
        file.open("productDetails.txt", ios::out);
```

```
Displaying details

Product name: wwerr
Product code: 12
Product quantity: 67

Product name: ver
Product code: 56
Product quantity: 98

Product name: ncnn
Product code: 38
Product quantity: 99

Product name: nndn
Product code: 37
Product quantity: 89

Product name: ddj
Product code: 90
Product quantity: 76
PS C:\Rohan\C and C++ programming lab>
```

```cpp
/*
INT105
CA3
Name: Rohan Kumar Saini
Roll No: RM2041A01
Reg No: 12011878
Question2
*/
#include <iostream>
using namespace std;

class Car
{
public:
    float horse_power, mileage;
    Car()
    {
        cout << "Enter mileage of the car: ";
        cin >> mileage;
        cout << "Enter horse power of the car: ";
        cin >> horse_power;
    }
    Car operator>(Car c)
    {
        if (mileage > c.mileage)
        {
            cout << "Mileage of car 1 is more" << endl;
        }
        else if (mileage == c.mileage)
        {
            cout << "Both cars have equal mileage" << endl;
        }
        else
        {
            cout << "Mileage of car 2 is more" << endl;
        }
        if (horse_power > c.horse_power)
        {
            cout << "Horse power of car 1 is more" << endl;
        }
        else if (horse_power == c.horse_power)
        {
            cout << "Both cars have same horse power" << endl;
        }
        else
        {
            cout << "Horse power of car 2 is more" << endl;
        }
    }
};

int main()
{
    cout << "Car 1" << endl;
    Car c1;
    cout << "Car 2" << endl;
    Car c2;
    cout << "\n";
    c1 > c2;
    return 0;
}
```

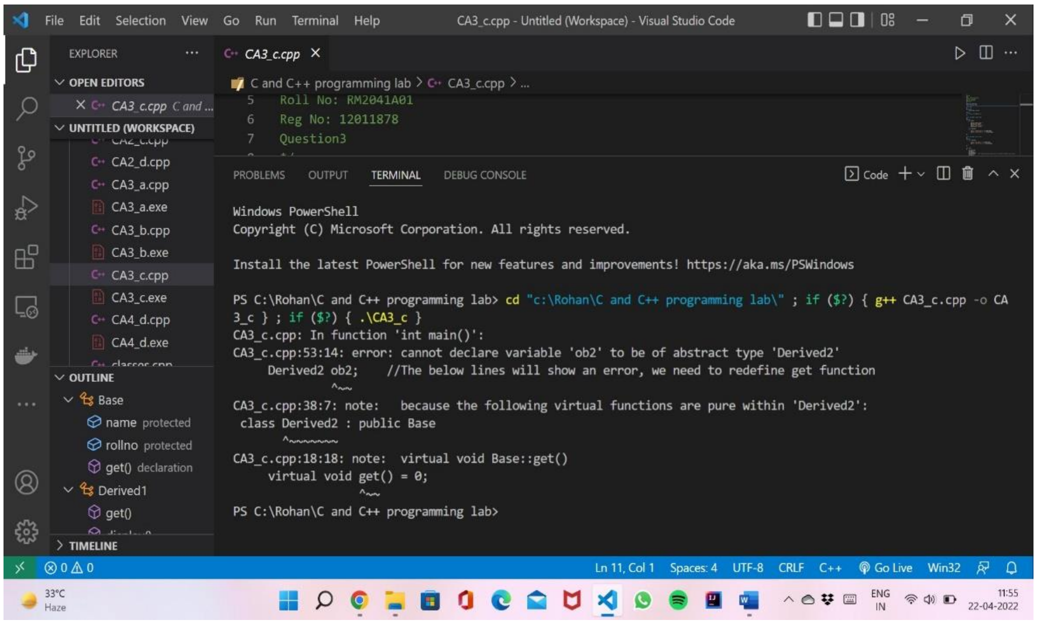


```
INT105
CA3
Name: Rohan Kumar Saini
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Rohan\C and C++ programming lab> cd "c:\Rohan\C and C++ programming lab\" ; if ($?) { g++ CA3_b.cpp -o CA
3_b } ; if ($?) { .\CA3_b }
Car 1
Enter mileage of the car: 55
Enter horse power of the car: 555
Car 2
Enter mileage of the car: 50
Enter horse power of the car: 560

Mileage of car 1 is more
Horse power of car 2 is more
PS C:\Rohan\C and C++ programming lab>
```

```
/*
INT105
CA3
Name: Rohan Kumar Saini
Roll No: RM2041A01
Reg No: 12011878
Question3
*/
#include <iostream>
using namespace std;

class Base
{
protected:
    string name, rollno;

public:
    virtual void get() = 0;
};

class Derived1 : public Base
{
public:
    void get()
    {
        cout << "Enter name: ";
        getline(cin, name);
        cout << "Enter roll no: ";
        getline(cin, rollno);
    }
    void display()
    {
        cout << "\nThe name is " << name<<endl;
        cout << "The roll no is " << rollno<<endl;
    }
};

class Derived2 : public Base
{
public:
    void display()
    {
        cout << "The name is " << name<<endl;
        cout << "The roll no is" << rollno<<endl;
    }
};

int main()
{
```

```cpp
    Derived1 ob1;
    ob1.get();
    ob1.display();
    Derived2 ob2;    //The below lines will show an error, we need
to redefine get function
    ob2.get();
    ob2.display();
    return 0;
}
```

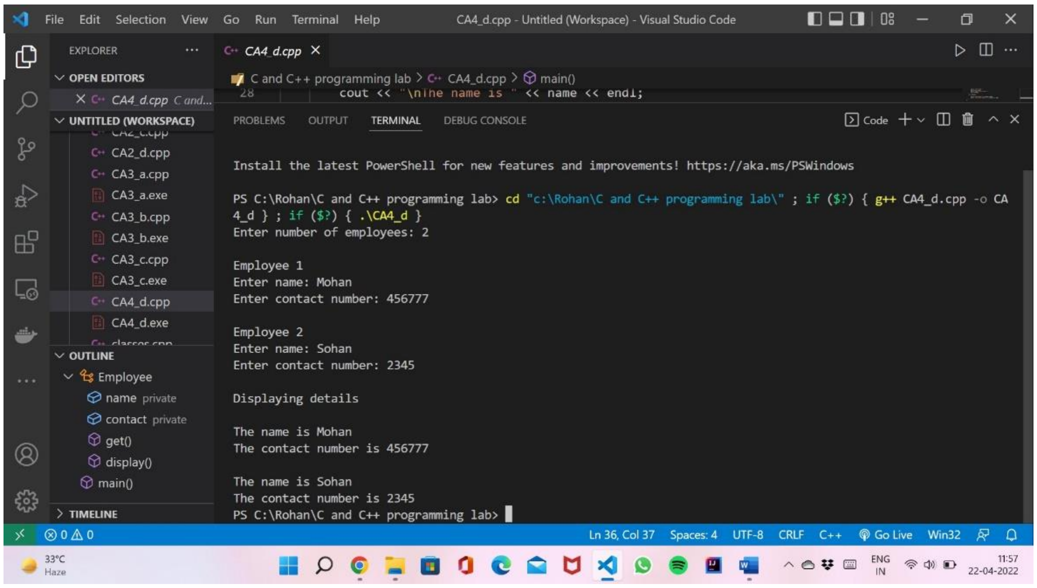


```
    Roll No: RM2041A01
    Reg No: 12011878
    Question3
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Rohan\C and C++ programming lab> cd "c:\Rohan\C and C++ programming lab\" ; if ($?) { g++ CA3_c.cpp -o CA
3_c } ; if ($?) { .\CA3_c }
CA3_c.cpp: In function 'int main()':
CA3_c.cpp:53:14: error: cannot declare variable 'ob2' to be of abstract type 'Derived2'
     Derived2 ob2;    //The below lines will show an error, we need to redefine get function
              ^~~
CA3_c.cpp:38:7: note:   because the following virtual functions are pure within 'Derived2':
 class Derived2 : public Base
       ^~~~~~~~
CA3_c.cpp:18:18: note:  virtual void Base::get()
     virtual void get() = 0;
                  ^~~
PS C:\Rohan\C and C++ programming lab>
```

```cpp
/*
INT105
CA3
Name: Rohan Kumar Saini
Roll No: RM2041A01
Reg No: 12011878
Question4
*/
#include <iostream>
using namespace std;

class Employee
{
    string name;
    long long int contact;

public:
    void get()
    {
        fflush(stdin);
        cout << "Enter name: ";
```

```cpp
        getline(cin, name);
        cout << "Enter contact number: ";
        cin >> contact;
    }
    void display()
    {
        cout << "\nThe name is " << name << endl;
        cout << "The contact number is " << contact << endl;
    }
};

int main()
{
    int n;
    cout << "Enter number of employees: ";
    cin >> n;
    Employee *p = new Employee[n];
    for (int i = 0; i < n; i++)
    {
        cout<<"\nEmployee "<<(i+1)<<endl;
        p[i].get();
    }
    cout<<"\nDisplaying details"<<endl;
    for (int i = 0; i < n; i++)
    {
        p[i].display();
    }
    delete[] p;
    return 0;
}
```



```
    cout << "\nThe name is " << name << endl;
```

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Rohan\C and C++ programming lab> cd "c:\Rohan\C and C++ programming lab\" ; if ($?) { g++ CA4_d.cpp -o CA
4_d } ; if ($?) { .\CA4_d }
Enter number of employees: 2

Employee 1
Enter name: Mohan
Enter contact number: 456777

Employee 2
Enter name: Sohan
Enter contact number: 2345

Displaying details

The name is Mohan
The contact number is 456777

The name is Sohan
The contact number is 2345
PS C:\Rohan\C and C++ programming lab> 
```