OBJECT ORIENTED PROGRAMMING SYSTEM
ASSIGNMENT-3

NAME – ATHARV DAYANAND SHET GOVEKAR

ROLL NO – 23B-CO-010

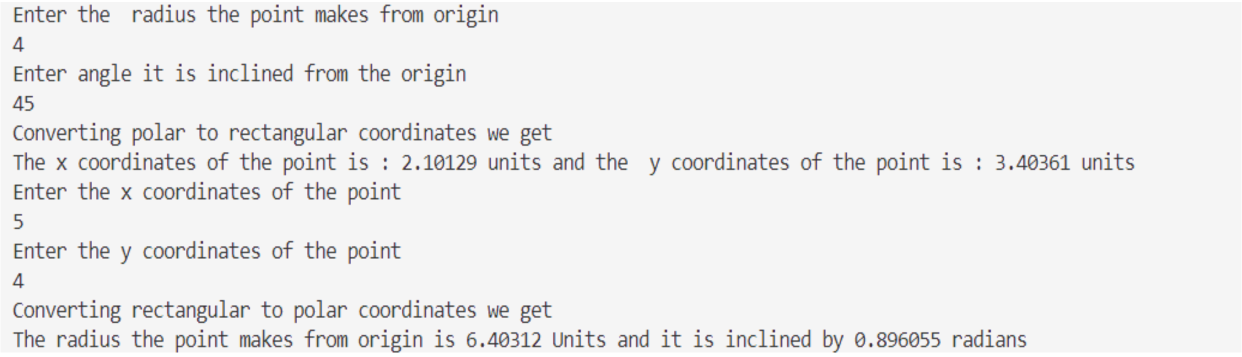
CLASS – SE COMP

1)

PROGRAM –

```
#include <iostream>

#include <cmath>

using namespace std;


class polar ;

class Rectangular {

float x ;

float y ;

public :

void getvalue(){

cout << "Enter the x coordinates of the point \n" ;

cin >> x ;

cout << "Enter the y coordinates of the point \n" ;

cin >> y ;

}

void display(){

   cout << "The x coordinates of the point is : " << x << "
units " <<

    "and the  y coordinates of the point is : " << y << "
units " <<  endl;

}

void friend polrtorect (Rectangular &a,polar &b);

void friend recttopolar (Rectangular &a,polar &b);

};

class polar {

   float radius ;

   float angle ;

   public :

 void getvalue (){

cout <<"Enter the  radius the point makes from
origin  \n" ;

cin>>radius ;

cout <<"Enter angle it is inclined from the origin \n" ;

cin>>angle;

}

void display (){

   cout<<"The radius the point makes from origin is
"<<radius<<" Units"<<" and it is inclined by "<<angle<<"
radians"<<endl;

}

void friend polrtorect (Rectangular &a,polar &b);

void friend recttopolar (Rectangular &a,polar &b);

};

void recttopolar (Rectangular &a,polar &b){

   float r,ang ;

   r = sqrt((a.x)*(a.x) + (a.y)*(a.y)) ;

   ang = atan(a.x/a.y) ;   //atan() function is used to
calculate tan inverse of a function

   b.radius = r ;

   b.angle = ang ;

}

void polrtorect (Rectangular &a,polar &b){

  a.x = (b.radius)*(cos(b.angle)) ;

  a.y = (b.radius)*(sin(b.angle)) ;

}

int main (){

polar P,Q;

Rectangular A,B ;

P.getvalue() ;

cout <<"Converting polar to rectangular coordinates we
get \n" ;

polrtorect(A,P) ;

A.display();

B.getvalue() ;

cout <<"Converting rectangular to polar coordinates we
get \n" ;

recttopolar(B,Q) ;

Q.display();

   return 0;}
```

OUTPUT

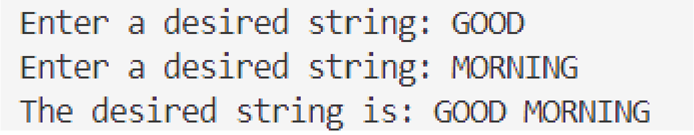


```
Enter the  radius the point makes from origin
4
Enter angle it is inclined from the origin
45
Converting polar to rectangular coordinates we get
The x coordinates of the point is : 2.10129 units and the  y coordinates of the point is : 3.40361 units
Enter the x coordinates of the point
5
Enter the y coordinates of the point
4
Converting rectangular to polar coordinates we get
The radius the point makes from origin is 6.40312 Units and it is inclined by 0.896055 radians
```

2) PROGRAM –

```
#include <iostream>

#include <cstring>

using namespace std;


class String {

private:

    char *p;

    int length;


public:

    // Default constructor

    String() {

        p = new char[1];

        p[0] = '\0';

        length = 0;

    }

    // Parameterized constructor

    String(const char *str) {

        length = strlen(str);

        p = new char[length + 1];

        strcpy(p, str);

    }

    // Concatenation of two String objects

    void join(const String &s1, const String &s2) {

        length = s1.length + s2.length;

        p = new char[length + 1];

        strcpy(p, s1.p);

        strcat(p, s2.p);

    }

    // Displaying the string

    void display()  {

        cout << "The desired string is: " << p << endl;

    }

    // Destructor to free allocated memory

    ~String() {

        delete[] p;

    }

};


int main() {

    String s1; //Default constructor called

    char s[20];

    cout << "Enter a desired string: ";

    cin.getline(s, 20);

    String s2(s); //Parameterized constructor called

    cout << "Enter a desired string: ";

    cin.getline(s, 20);

    String s3(s); //Parameterized constructor called

    String s4 ;

    s4.join(s2, s3);

    s4.display();

    // Destructors will be called automatically here when objects go out of scope

    return 0;}
```

OUTPUT

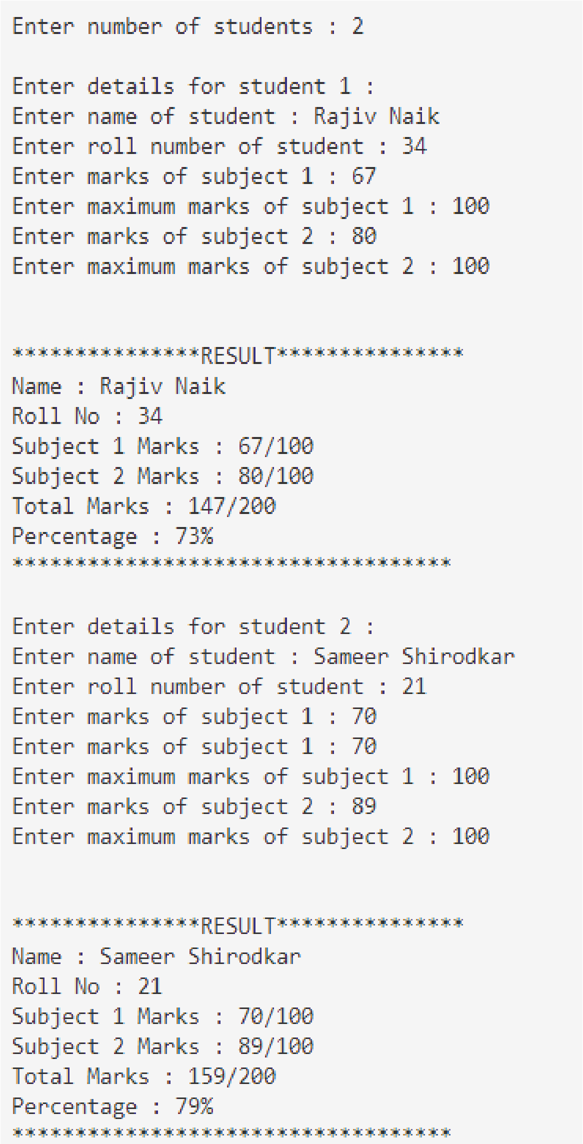


```
Enter a desired string: GOOD
Enter a desired string: MORNING
The desired string is: GOOD MORNING
```

3) PROGRAM

```
#include <iostream>

#include <cstring>

using namespace std ;


class student {

char *name ;

int roll_no ;

public :

student (){

name = new char[1] ;

name[0] = '\0' ;

roll_no = 0 ;

}

student (char *n , int r) {

name = new char[strlen(n)+1] ;

strcpy(name , n) ;

roll_no = r ;

}

void display (){

   cout << "Name : " << name << endl ;

   cout << "Roll No : " << roll_no << endl ;

}

~student (){

   delete[] name ;

}};


class test : public student {

int sub1_marks , sub2_marks ;

int sub1_max , sub2_max ;

public :

test () : student() {

sub1_marks = 0 ;

sub2_marks = 0 ;

sub1_max = 100 ;

sub2_max = 100 ;

}

test (char *n , int r , int s1 , int s2, int max1, int max2) : student(n , r) {

sub1_marks = s1 ;

sub2_marks = s2 ;

sub1_max = max1 ;

sub2_max = max2 ;

}

void display (){

   student::display() ;

   cout << "Subject 1 Marks : " << sub1_marks <<"/"<<sub1_max<< endl ;

   cout << "Subject 2 Marks : " << sub2_marks <<"/"<<sub2_max << endl ;

}

~test (){

}

};


class result : public test {

int total ;

int max ;

float percentage ;

public :

result () : test() {

   total = 0 ;

   max = 200 ;

   percentage = 0 ;

}

result (char *n , int r , int s1 , int s2 , int max1 ,int max2 ) : test(n , r , s1 , s2, max1, max2) {

   total = s1 + s2 ;
   max = max1 + max2 ;
   percentage = (total*100)/max ;
}
void display (){
   cout <<
"\n\n***************RESULT***************" <<
endl ;
   test::display() ;
   cout << "Total Marks : " << total<<"/"<<max<< endl ;
   cout << "Percentage : " << percentage << "%" << endl ;
   cout << "***********************************"
<< endl ;
}
~result (){
}
};


int main() {
   cout <<"\n\nEnter number of students : " ;
   int num ;
   cin >> num ;
   for (int i = 0 ; i < num ; i++) {
      cout << "\nEnter details for student " << i+1 << " : "
<< endl ;
      char n[20] ;
      int r , s1 , s2, max1 , max2 ;
      cout << "Enter name of student : " ;
      cin.ignore() ;
      cin.getline(n , 20) ;
      cout << "Enter roll number of student : " ;
      cin >> r ;
      cout << "Enter marks of subject 1 : " ;
      cin >> s1 ;
      cout << "Enter maximum marks of subject 1 : " ;
      cin >> max1 ;
      cout << "Enter marks of subject 2 : " ;
      cin >> s2 ;
      cout << "Enter maximum marks of subject 2 : " ;
      cin >> max2 ;
      result r1(n , r , s1 , s2 ,max1 , max2) ;
      r1.display() ;
   }
   return 0;
}
```

OUTPUT –



```
Enter number of students : 2

Enter details for student 1 :
Enter name of student : Rajiv Naik
Enter roll number of student : 34
Enter marks of subject 1 : 67
Enter maximum marks of subject 1 : 100
Enter marks of subject 2 : 80
Enter maximum marks of subject 2 : 100


***************RESULT***************
Name : Rajiv Naik
Roll No : 34
Subject 1 Marks : 67/100
Subject 2 Marks : 80/100
Total Marks : 147/200
Percentage : 73%
***********************************

Enter details for student 2 :
Enter name of student : Sameer Shirodkar
Enter roll number of student : 21
Enter marks of subject 1 : 70
Enter marks of subject 1 : 70
Enter maximum marks of subject 1 : 100
Enter marks of subject 2 : 89
Enter maximum marks of subject 2 : 100


***************RESULT***************
Name : Sameer Shirodkar
Roll No : 21
Subject 1 Marks : 70/100
Subject 2 Marks : 89/100
Total Marks : 159/200
Percentage : 79%
***********************************
```

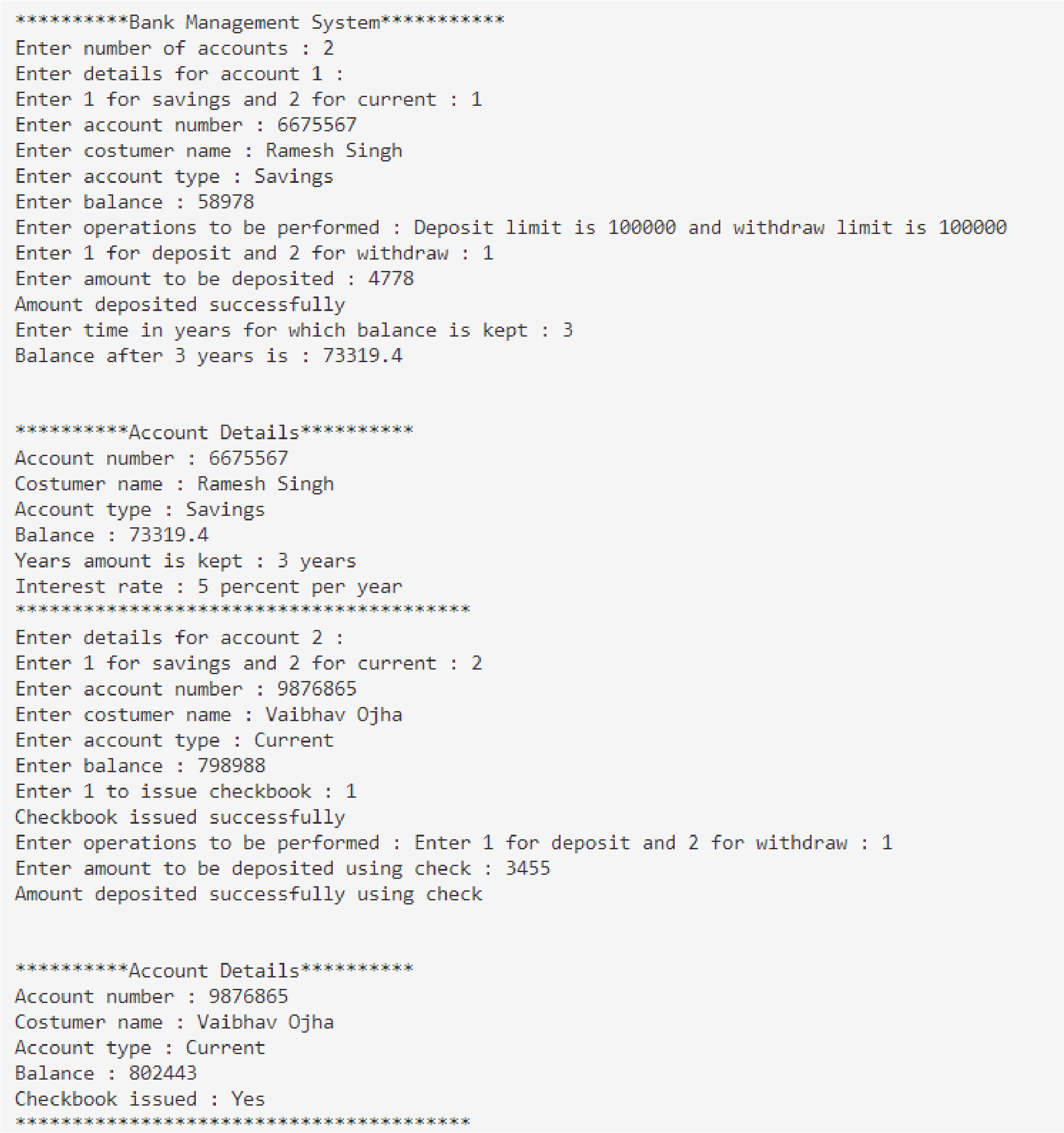
4) PROGRAM –

```
#include <iostream>

#include <cstring>

using namespace std ;


class account {

   protected :

long int account_no ;

char *costumer_name ;

char *account_type ;

float balance ;


public :

void get_info (){

   cout << "Enter account number : " ;

   cin >> account_no ;

   cout << "Enter costumer name : " ;

   char n[20] ;

   cin.ignore() ;

   cin.getline(n , 20) ;

   costumer_name = new char[strlen(n)+1] ;

   strcpy(costumer_name , n) ;

   cout << "Enter account type : " ;

   char t[20] ;

   cin.getline(t , 20) ;

   account_type = new char[strlen(t)+1] ;

   strcpy(account_type , t) ;

   cout << "Enter balance : " ;

   cin >> balance ;

}

void display (){

   cout << "Account number : " << account_no << endl ;

   cout << "Costumer name : " << costumer_name <<
endl ;

   cout << "Account type : " << account_type << endl ;

   cout << "Balance : " << balance << endl ;

}

void show_balance (){

  cout << "The bank account has a balance of Rs." <<
balance << endl ;


}

};


class savings : public account {


float interest_rate = 0.05 ;

int withdraw ;

int withdraw_limit = 100000 ;

int deposit ;

float time ;

int deposit_limit = 100000 ;


public :

void showinterest_rate (){

   cout << "The interest rate is " << interest_rate << endl
;}


 void oper (){


   cout << "Enter operations to be performed : " ;

   cout<<"Deposit limit is "<<deposit_limit<<" and
withdraw limit is "<<withdraw_limit<<endl;

   cout << "Enter 1 for deposit and 2 for withdraw : " ;

   int choice ;

   cin >> choice ;

   if (choice == 1) {

      cout << "Enter amount to be deposited : " ;

      cin >> deposit ;

      if (deposit > deposit_limit) {
         cout << "Deposit limit exceeded" << endl ;
      }
      else {
         balance += deposit ;
         cout << "Amount deposited successfully" << endl
;
      }
   }
   else if (choice == 2) {
      cout << "Enter amount to be withdrawn : " ;
      cin >> withdraw ;
      if (withdraw > withdraw_limit) {
         cout << "Withdraw limit exceeded" << endl ;
      }
      else {
         balance -= withdraw ;
         cout << "Amount withdrawn successfully" <<
endl ;
      }
   }
   else {
      cout << "Invalid choice" << endl ;
   }
if(balance < 500 ){
   balance = balance - 100 ;
}
 }
void keep_balance (){
   cout << "Enter time in years for which balance is kept
: " ;
   cin >> time ;
   balance += balance * interest_rate * time ;
   cout << "Balance after " << time << " years is : " <<
balance << endl ;
}
void display (){
```

```
cout <<"\n\n**********Account
Details**********"<<endl;
account::display() ;
   cout << "Years amount is kept : " << time << " years"
<< endl ;
   cout << "Interest rate : " <<(interest_rate*100) <<"
percent per year"<< endl ;
 cout <<
"****************************************" <<
endl ;
  }
};


class current : public account {
   bool checkbook ;
   int issue ;
   int withdraw ;
   int deposit ;
  public :
  void issuecheckbook (){
       cout << "Enter 1 to issue checkbook : " ;
    cin>>checkbook ;
issue = 1 ;
     cout << "Checkbook issued successfully" << endl ;
  }
void oper (){
   cout << "Enter operations to be performed : " ;
   cout << "Enter 1 for deposit and 2 for withdraw : " ;
   int choice ;
   cin >> choice ;
   if (choice == 1) {
      cout << "Enter amount to be deposited using check
: " ;
      cin >> deposit ;
      balance += deposit ;
      cout << "Amount deposited successfully using
check" << endl ;
   }
```

```
    else if (choice == 2) {
        cout << "Enter amount to be withdrawn using check : " ;
        cin >> withdraw ;
        balance -= withdraw ;
        cout << "Amount withdrawn successfully using check" << endl ;
    }
    else {
        cout << "Invalid choice" << endl ;
    }
    if(balance < 500 ){
    balance = balance - 100 ;
}
}
void display (){
    cout <<"\n\n**********Account Details**********"<<endl;
    account::display() ;
 if (issue==1){
     cout << "Checkbook issued : Yes" << endl ;
 }
 else {
     cout << "Checkbook issued : No" << endl ;
 }
  cout << "****************************************" << endl ;
}
};


int main() {
    int num,choice ;
cout<<"**********Bank Management System***********"<<endl;
cout<<"Enter number of accounts : " ;
cin >> num ;
for(int i = 0 ; i < num ; i++) {
    cout << "Enter details for account " << i+1 << " : " << endl ;
    cout << "Enter 1 for savings and 2 for current : " ;
    cin >> choice ;
    if (choice == 1) {
        savings s ;
        s.get_info() ;
        s.oper() ;
        s.keep_balance() ;
        s.display() ;
    }
    else if (choice == 2) {
        current c ;
        c.get_info() ;
        c.issuecheckbook() ;
        c.oper() ;
        c.display() ;
    }
    else {
        cout << "Invalid choice" << endl ;
    }
}

    return 0;
}
```

OUTPUT –

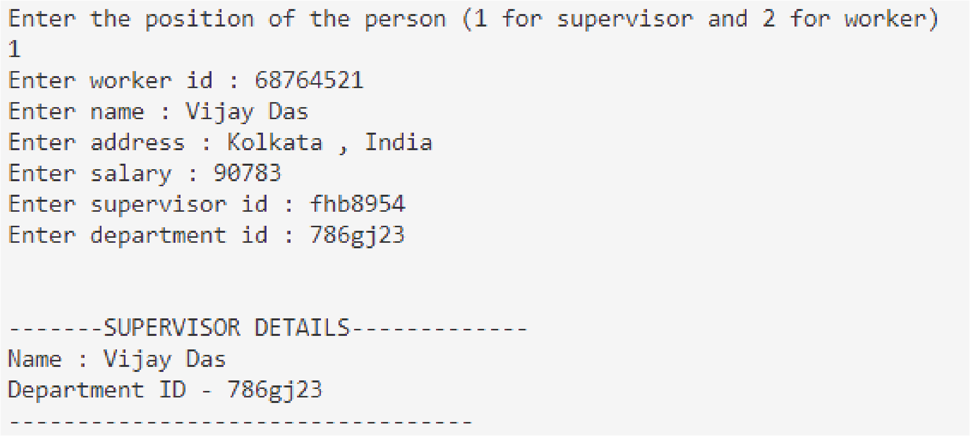


```
**********Bank Management System***********
Enter number of accounts : 2
Enter details for account 1 :
Enter 1 for savings and 2 for current : 1
Enter account number : 6675567
Enter costumer name : Ramesh Singh
Enter account type : Savings
Enter balance : 58978
Enter operations to be performed : Deposit limit is 100000 and withdraw limit is 100000
Enter 1 for deposit and 2 for withdraw : 1
Enter amount to be deposited : 4778
Amount deposited successfully
Enter time in years for which balance is kept : 3
Balance after 3 years is : 73319.4


**********Account Details**********
Account number : 6675567
Costumer name : Ramesh Singh
Account type : Savings
Balance : 73319.4
Years amount is kept : 3 years
Interest rate : 5 percent per year
****************************************
Enter details for account 2 :
Enter 1 for savings and 2 for current : 2
Enter account number : 9876865
Enter costumer name : Vaibhav Ojha
Enter account type : Current
Enter balance : 798988
Enter 1 to issue checkbook : 1
Checkbook issued successfully
Enter operations to be performed : Enter 1 for deposit and 2 for withdraw : 1
Enter amount to be deposited using check : 3455
Amount deposited successfully using check


**********Account Details**********
Account number : 9876865
Costumer name : Vaibhav Ojha
Account type : Current
Balance : 802443
Checkbook issued : Yes
****************************************
```

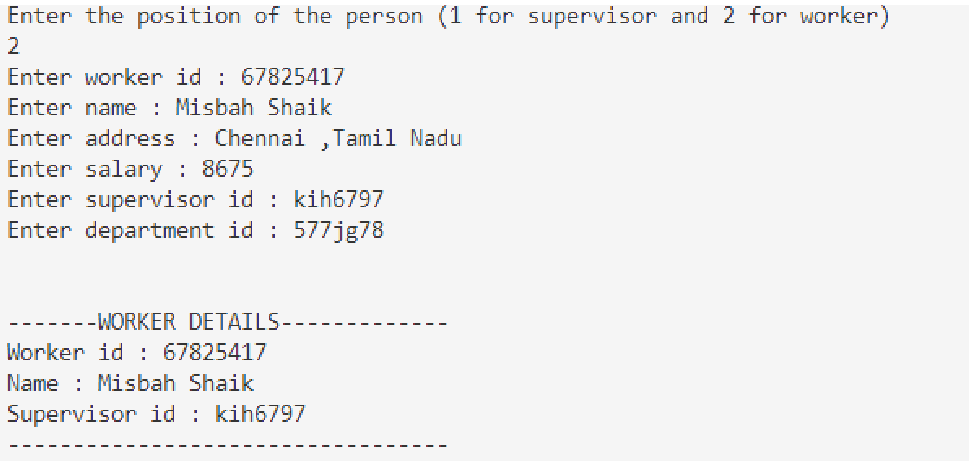
5) PROGRAM –

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

class worker {
  protected :
  long int worker_id ;
  char *name ;
  char *address ;
  float salary ;
  char *supervisor_id ;
  char *department_id ;

  public :
  worker (){
    worker_id = 0 ;
    name = new char[20] ;
    address = new char[20] ;
    salary = 0 ;
    supervisor_id = new char[20] ;
    department_id = new char[20] ;
  }
  void get_info (){
    cout << "Enter worker id : " ;
   cin >> worker_id ;
    cout << "Enter name : " ;
    char n[50] ;
    cin.ignore() ;
    cin.getline(n , 50) ;
    name = new char[strlen(n)+1] ;
    strcpy(name , n) ;
    cout << "Enter address : " ;
    char a[50] ;
    cin.getline(a , 50) ;
    address = new char[strlen(a)+1] ;
    strcpy(address , a) ;
    cout << "Enter salary : " ;
    cin >> salary ;
    cout << "Enter supervisor id : " ;
    char s[50] ;
    cin.ignore() ;
    cin.getline(s , 50) ;
    supervisor_id = new char[strlen(s)+1] ;
    strcpy(supervisor_id , s) ;
    cout << "Enter department id : " ;
    char d[50] ;
    cin.getline(d , 50) ;
    department_id = new char[strlen(d)+1] ;
    strcpy(department_id , d) ;
}
virtual void display (){
  cout <<"\n\n-------WORKER DETAILS-------------\n" ;
  cout << "Worker id : " << worker_id << endl ;
  cout << "Name : " << name << endl ;
  cout << "Supervisor id : " << supervisor_id << endl ;
  cout<<"----------------------------------\n\n";
}


};

class supervisor :public worker {
public :
void display (){
  cout <<"\n\n-------SUPERVISOR DETAILS-------------\n" ;
  cout << "Name : "<< name <<endl ;
  cout << "Department ID "<<department_id << endl ;
  cout<<"----------------------------------\n\n";
}
};

int main(){
   worker *bptr ;
   int choice;

   cout << "Enter the position of the person"
  <<" (1 for supervisor and 2 for worker)"<<endl ;
cin>>choice ;
if (choice==1){
   supervisor s ;
   bptr =&s ;
   bptr->get_info ();
   bptr->display() ;
}
else if (choice ==2){
   worker w ;
   bptr = &w ;
   bptr->get_info();
   bptr->display() ;
}
else {
   cout<<"Enter valid choice value\n";
}
return 0 ;
}
```

OUTPUT –



```
Enter the position of the person (1 for supervisor and 2 for worker)
1
Enter worker id : 68764521
Enter name : Vijay Das
Enter address : Kolkata , India
Enter salary : 90783
Enter supervisor id : fhb8954
Enter department id : 786gj23


-------SUPERVISOR DETAILS-------------
Name : Vijay Das
Department ID - 786gj23
----------------------------------
```



```
Enter the position of the person (1 for supervisor and 2 for worker)
2
Enter worker id : 67825417
Enter name : Misbah Shaik
Enter address : Chennai ,Tamil Nadu
Enter salary : 8675
Enter supervisor id : kih6797
Enter department id : 577jg78


-------WORKER DETAILS-------------
Worker id : 67825417
Name : Misbah Shaik
Supervisor id : kih6797
---------------------------------
```