Exercise 1 – Inheritance

1. Create a class called Employee

2. Employee class should have the properties called - empNo, name, designation, and basicSalary 3. Implement the overloaded constructor.

4. Implement the method called ***display()*** in the Employee class. This method should display the properties of the Employee class.

5. By calling the base class constructor create a dynamic object of the Employee class. Assign 1003, Tharindu, General Manager, 75000 for the variables empNo, name, designation and basicSalary.

6. Call ***display()*** method for the Employee class object.

7. Implement the AcademicStaff class derived from the Employee class which has the property of allowance.

8. Implement the ***display()*** method to show all the properties of the Employee class and AcademicStaff class (Use method overriding)

9. Implement the NonAcademic class derived from the Employee class which has the properties of otRate and otHours

10. Implement the ***display()*** method to show all the properties of the Employee class and NonAcademic class (Use method overriding)

11. Implement the virtual method called ***calcNetSalary()*** in the Employee class.

12. Implement ***calcNetSalary()*** in AcademicStaff and NonAcademic classes to calculate the net salary.

main.cpp

```
#include <iostream>
#include "Employee.h"

int main() {
  //Create a dynamic object of the Employee class.

  // Don't change the code below
   AcademicStaff *emp1;
  emp = new AcademicStaff(1001, "Kusal", "Lecturer", 100000, 25000);
  NonAcademic emp2(1002, "Janith", "Manager", 60000, 2500, 20);
  emp1->display();
  emp1->calcNetSalary();
  emp2.display();
  emp2.calcNetSalary();
}
```

Employee.h

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

//Base class
class Employee{
 protected:
 //Declare the variables

 public:
  //Constructor
  Employee(int pempNo, char pname[], char pdesignation[], double pbasicSalary);

  //Declare methods
  void display();
};

class AcademicStaff: public Employee{
 protected:
 //Declare the variables

 public:
 //Constructor
 AcademicStaff(int pempNo, char pname[], char pdesignation[], double pbasicSalary, double
pallowance);

 //Declare Methods
 void display();
}

class NonAcademic: public Employee{
 protected:
 //Declare the variables

 public:
 //Constructor
 NonAcademic(int pempNo, char pname[], char pdesignation[], double pbasicSalary, double potRate,
float potHours);

 //Declare Methods
 void display();
}
```

Employee.cpp

```
#include "Employee.h"
#include <iostream>
using namespace std;

//================Employee Class=================================
//Constructor
Employee::Employee(int pempNo, char pname[], char pdesignation[], double pbasicSalary){
  // Implementation

}
//Methods of Employee Class



//==================AcademicStaff Class=================================
AcademicStaff::AcademicStaff(int pempNo, char pname[], char pdesignation[], double pbasicSalary,
double pallowance):Employee(pempNo, pname[], pdesignation[], pbasicSalary){
  // rest of the implementation

}

//Methods of the AcademicStaff
void AcademicStaff:: display(){
  Employee::display(); // This calls the display() method of the base class to display the properties of
the base class.
  //Rest of the implementation
}


//=========================NonAcademic Class=========================
NonAcademic::NonAcademic(int pempNo, char pname[], char pdesignation[], double pbasicSalary,
double potRate, float potHours):Employee(pempNo, pname[], pdesignation[], pbasicSalary){
   // rest of the implementation

}

//Methods of the NonAcademic Class
```