
Tutorial 8

Inheritance

Program 1

Overloaded vs. Redefined Methods

Determine the output printed by the following program.

```
1  #include<iostream>
2
3  using namespace std;
4
5  class Parent{
6  public:
7      Parent(){}
8      void print(){ cout << "Print from Parent" << endl; }
9  };
10
11 class Child : public Parent{
12 public:
13     Child(){}
14
15     void print(){ cout << "Print from Child" << endl;} // Redefined
16
17     void print(string name){ cout << "Child prints name " << name << endl; } // Overloaded
18
19     void print(int number){ cout << "Child prints number " << number << endl; } // Overloaded
20
21     void print(int n1, int n2){ cout << "Child prints two numbers, " << n1 << " and " << n2 << endl; } // Overloaded
22 };
23
24 int main()
25 {
26     Parent p;
27     Child c;
28
29     p.print();
30     c.print();
31     c.print(10);
32     c.print("Ahmad");
33     c.print(88,99);
34
35     return 0;
36 }
37
```

Program 2

Redefined Methods (cont.)

As for this problem, you will be doing an active learning activity as follows:

1. Read the question
2. Think the answer. NO TALKING.
3. Choose your answer and submit it on elearning.
4. Debate your answer with others. Move around.
5. Revise your answer (if needed)

Question:

Determine the output printed by the following program. Choose the answer A, B, C, or D and submit on elearning.

```
1  #include <iostream>
2
3  using namespace std;
4
5  class Shape
6  {
7  public:
8      Shape() {}
9      int getArea() const { return 0; }
10
11     void print()
12     {
13         int a = getArea();
14         cout << "Area = " << a << endl;
15     }
16 };
17
18 class Rectangle : public Shape
19 {
20 private:
21     int width, length;
22
23 public:
24     Rectangle(int w, int l)
25     {
26         width = w;
27         length = l;
28     }
29
30     int getArea() const { return width * length; }
31 };
32
33 int main()
34 {
35     Shape s;
36     Rectangle r(2, 6);
37
38     s.print();
39     r.print();
40
41     system("pause");
42     return 0;
43 }
```

A.
Area = 0
Area = 12

B.
Area = 12
Area = 0

C.
Area = 0
Area = 0

D.
Area = 12
Area = 12

Program 3

You will be writing a program to record student information of a university. There are two types of students, **undergraduate** and **postgraduate** students. The program needs to record the **name** and **matriculation** number for each student. Besides, the program also needs to record the **cgpa** for each undergraduate student, and the **project** information for each postgraduate student. Regarding the project of the postgraduate students, each project consists of a **title** and **area** or discipline.

Tasks

The exercise is divided into three parts:

- Part 1 is about design. You will be doing this part on a piece of paper and it needs to be handed in.
- Part 2 consists of guided tasks, in which the questions or tasks are accompanied with the code segment for the solution. This part will not be assessed. However, this part is a pre-requisite for the next part.
- Part 3 consists of tasks that you need to do it yourself.

Part 1: Design

1. Analyze the given case study and draw the **class diagram** consisting of classes **Student**, **Project**, **Undergraduate**, and **Postgraduate**, as well as their relationships.

Part 2: Guided Coding Tasks

2. Full implementations for the classes `Student` and `Project` have been given in the template program **program3.cpp**. Review the code of these classes.
3. Complete the implementation for the class `Undergraduate`. Here, you are supposed to implement inheritance. The solution is given in the following code segment. Copy only the highlighted lines from the code segment below. Then review the code.

```
class Undergraduate : public Student
{
protected:
    double cgpa;

public:
    Undergraduate( string _name = "",
                  string _matric = "",
                  double _cgpa = 0.0 ) : Student(_name, _matric), cgpa(_cgpa)
    {}

    double getCGPA() const { return cgpa; }
    void setCGPA(double c) { cgpa = c; }

    void input()
    {
        Student::input();
        cout << "Enter CGPA => ";
        cin >> cgpa;
    }
}
```

```

void print() const
{
    Student::print();
    cout << "CGPA : " << cgpa << endl;
}
};

```

Part 3: Graded Coding Tasks

4. Complete the implementation for the class `Postgraduate`.

Task 5 to 8 below are related to the main function:

5. Create two arrays to hold lists of undergraduate and postgraduate students, respectively.
6. Read data for an undergraduate student and add it to the relevant array.
7. Read data for a postgraduate student and add it to the relevant array.
8. Display the lists of undergraduate and postgraduate students.

Figure 3.1 (a) to (g) shows expected result that the program should produce. The bold texts are user inputs.

```

===== [ Menu ] =====
1. Add an undergraduate student
2. Add a postgraduate student
3. Display all students
9. Exit

Choose an operation [1,2,3 or 9] => 1

Adding an undergraduate student:

Enter name => Aminah Abdullah
Enter matric => A16CS5022
Enter CGPA => 3.85

```

(a) The user has chosen operation 1 to add a new undergraduate student

```

===== [ Menu ] =====
1. Add an undergraduate student
2. Add a postgraduate student
3. Display all students
9. Exit

Choose an operation [1,2,3 or 9] => 1

```

Adding an undergraduate student:

Enter name => **Abdul Jalil Rahman**

Enter matric => **B17CS3012**

Enter CGPA => **3.7**

(b). The user has chosen operation 1 to add a new undergraduate student

===== [Menu] =====

1. Add an undergraduate student

2. Add a postgraduate student

3. Display all students

9. Exit

Choose an operation [1,2,3 or 9] => **2**

Adding a postgraduate student:

Enter name => **Hassan Shahir**

Enter matric => **PC150523**

Enter project title => **Machine learning with abnormal data**

Enter project area => **Artificial Intelligence**

(c). The user has chosen operation 2 to add a postgraduate student

===== [Menu] =====

1. Add an undergraduate student

2. Add a postgraduate student

3. Display all students

9. Exit

Choose an operation [1,2,3 or 9] => **2**

Adding a postgraduate student:

Enter name => **Siti Nurliyana Kamaruddin**

Enter matric => **MA185246**

Enter project title => **Cost-effective house building**

Enter project area => **Building and construction**

(d). The user has chosen operation 2 to add a postgraduate student

===== [Menu] =====

1. Add an undergraduate student

2. Add a postgraduate student

3. Display all students

9. Exit

Choose an operation [1,2,3 or 9] => **1**

Adding an undergraduate student:

Enter name => **Zainuddin Abdul Jalil**

Enter matric => **B18CS4586**

Enter CGPA => 3

(e). The user has chosen operation 1 to add a new undergraduate student

=====[Menu]=====

1. Add an undergraduate student

2. Add a postgraduate student

3. Display all students

9. Exit

Choose an operation [1,2,3 or 9] => **3**

The list of undergraduate students:

No. : 1

Name : Aminah Abdullah

Matric: A16CS5022

CGPA : 3.85

No. : 2

Name : Abdul Jalil Rahman

Matric: B17CS3012

CGPA : 3.7

No. : 3

Name : Zainuddin Abdul Jalil

Matric: B18CS4586

CGPA : 3

The list of postgraduate students:

No. : 1

Name : Hassan Shahir

Matric: PC150523

Project title : Machine learning with abnormal data

Project area : Artificial Intelligence

No. : 2

Name : Siti Nurliyana Kamaruddin

Matric: MA185246

Project title : Cost-effective house building

Project area : Building and construction

(f). The user has chosen operation 3 to list the students

```
===== [ Menu ] =====
```

1. Add an undergraduate student
2. Add a postgraduate student
3. Display all students
9. Exit

```
Choose an operation [1,2,3 or 9] => 9
```

(g). The user has chosen operation 9 to end the program

Figure 3.1

Program 4

Faculty of Engineering will be organizing a competition that is open to all UTM students. The competition is meant to provide a platform to the students to exhibit their idea. Some important notes regarding the competition:

- It is an individual participation.
- Each student must have a project.
- Each student has to find an advisor which can be any of UTM staff regardless of their position. That means, the adviser can a lecturer, technician, research officer. However, only one advisor is allowed for each student.
- Also, a staff can only advice one student.

You are hired by the faculty to develop a program to manage the records of the participants. Suppose the class design has been done by other parties as shown in Figure 4.1. You are also provided an incomplete program (**program4.cpp**). Complete the program based on the tasks stated in the program (Task 1 to Task 16). The program should produce output like shown in Figure 4.2.

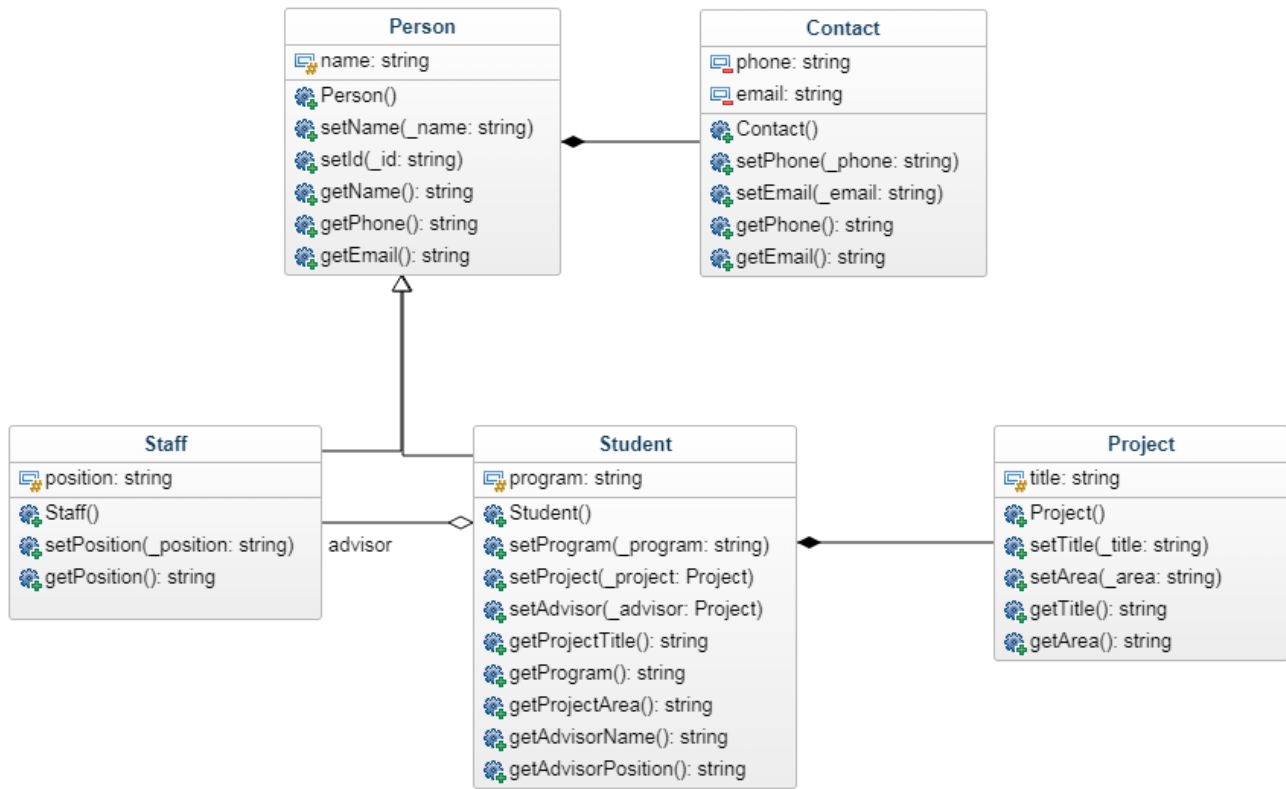


Figure 4.1: The class diagram

```

----- First Printing -----

Student # 1
Name           : Lim Sew Ying
Phone          : 010-897634045
Email          : lsying1212@live.utm.my
Project Title  : Smart Dustbin
Advisor's Name : 
Advisor's Position : 

Student # 2
Name           : Abu Bakar Razali
Phone          : 011-1210000
Email          : abraz78@live.utm.my
Project Title  : Money Recognition for Blind People
Advisor's Name : 
Advisor's Position : 

Student # 3
Name           : Nur Amalina Muhammad
Phone          : 012-9000123
Email          : nama978@live.utm.my
Project Title  : Dengue Testing Kit
Advisor's Name : 
Advisor's Position : 

----- Second Printing -----
  
```



```

Student # 1
Name           : Lim Sew Ying
Phone          : 010-897634045
Email          : lsying1212@live.utm.my
Project Title  : Smart Dustbin
Advisor's Name : Dr. Siti Zubaidah Rosli
Advisor's Position : Research Officer

Student # 2
Name           : Abu Bakar Razali
Phone          : 011-1210000
Email          : abraz78@live.utm.my
Project Title  : Money Recognition for Blind People
Advisor's Name : Mr. Qamarool Zaman
Advisor's Position : Senior Technician

Student # 3
Name           : Nur Amalina Muhammad
Phone          : 012-9000123
Email          : nama978@live.utm.my
Project Title  : Dengue Testing Kit
Advisor's Name : Prof. Dr. Muhammad Roslan Abdullah
Advisor's Position : Lecturer

```

----- Third Printing -----

```

Student # 1
Name           : Lim Sew Ying
Phone          : 010-897634045
Email          : lsying1212@live.utm.my
Project Title  : Smart Dustbin
Advisor's Name : Mr. Qamarool Zaman
Advisor's Position : Senior Technician

Student # 2
Name           : Abu Bakar Razali
Phone          : 011-1210000
Email          : abraz78@live.utm.my
Project Title  : Money Recognition for Blind People
Advisor's Name : Dr. Siti Zubaidah Rosli
Advisor's Position : Research Officer

Student # 3
Name           : Nur Amalina Muhammad
Phone          : 012-9000123
Email          : nama978@live.utm.my
Project Title  : Dengue Testing Kit
Advisor's Name :
Advisor's Position :

```

Figure 4.2: Expected output

Program 5

Consider the program in program5.cpp.

- What's wrong with the code?
- Then, fix the problem by refactoring the code using **inheritance**.
- Also, draw the class diagram for the refactored program.

Notes: the class diagram of the original program should look like below:

