University of Lincoln
School of Computer Science
CMP9133M – Advanced Programming
Workshop 10

Task(Assessed): University Management System

Design a University Management System that models the hierarchy of students, faculty, and staff. Implement the system using classes and demonstrate the use of **inheritance**, **polymorphism**, and **memory management** techniques.

Instructions:

1. Create a base class Person with private attributes:

    - std::string name
    - int age

    Implement public member functions for getting and setting the name and age.

2. Derive a class Student from the Person class, with additional private attributes:

    - std::string studentId
    - double gpa

    Implement public member functions for getting and setting the student ID and GPA.

3. Derive a class Faculty from the Person class, with additional private attributes:

    - std::string facultyId
    - std::string department

    Implement public member functions for getting and setting the faculty ID and department.

4. Derive a class Staff from the Person class, with additional private attribute:

    - std::string staffId

    Implement public member functions for getting and setting the staff ID.

5. In the main() function:

    - Dynamically allocate memory for an array of pointers to Person objects of size 5.
    - Prompt the user to enter details for each person (name, age, and type - student, faculty, or staff).
    - Based on the user's input, create the respective objects (student, faculty, or staff) dynamically using new and store the object in the array.
    - Iterate through the array and display the details of each person using polymorphism.
    - Deallocate the memory for each object using delete.
    - Deallocate the memory for the array.

Expected output:

```
Enter the details of Person 1:
Name: John Doe
Age: 20
Enter the type of person (S for Student, F for Faculty, ST for Staff): F
Enter the Faculty ID: F101
Enter the Department: Computer Science

Enter the details of Person 2:
Name: Jane Smith
Age: 25
Enter the type of person (S for Student, F for Faculty, ST for Staff): S
Enter the Student ID: S102
Enter the GPA: 3.8

Enter the details of Person 3:
...
```

Note: Remember to handle dynamic memory allocation and deallocation properly, and perform input validation where necessary.