National University of Singapore
School of Continuing and Lifelong Education

TIC1002: Introduction to Computing and Programming II
Semester II, 2020/2021

Tutorial 7
Object Oriented Programming

1. Given the following five functions:

```
// Function 1
int add(int a, int b) { return a + b; }

// Function 2
double add(double a, double b) { return a + b; }

// Function 3
double add(int a, int b) { return a + b; }

// Function 4
void swap(int a, int b) {
  a = a ^ b;
  b = a ^ b;
  a = a ^ b;
}

// Function 5
void swap(int& a, int& b) {
  a = a ^ b;
  b = a ^ b;
  a = a ^ b;
}
```

   (a) Which functions are having conflict in the overloading?

   (b) What causes those overloading errors?

   (c) Removing the offending function, would the following statement cause error?

   If yes, how would you correct the error (besides removing the offending function call)?

   If no, which function is called in each statement?

```
add(1, 2);
add(3.0, 4);
add(5, 6.0);
add(7.0, 8.0);
```

2. Identify and correct the errors in the code below:

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

class Person {
private:
    string name;
    int age;
    Person mother;

public:
    Person(string name, int age) {
        name = name;
        age = age;
    }

    Person(string name, int age, Person person) {
        name = name;
        age = age;
        mother = &person;
    }

    string get_name() { return name; }
    int get_age() { return age; }
    Person get_mother() { return mother; }
}

int main() {
    Person a;
    Person sally("Sally", 32);
    Person thomas("Thomas", 6, sally);
    cout << sally.name << " is the mother of " << thomas.name << endl;
    cout << "their age difference is " << sally.age - thomas.age
         << " years" << endl;
}
```

3. You might have heard the nursery rhyme "Old McDonald". Its lyrics teach kids about the sounds animals make. You may recall the part of the lyrics about the *Cow*:

   Old McDonald had a farm, E-I-E-I-O
   And on his farm he had a *Cow*, E-I-E-I-O
   With a **Moo Moo** here and a **Moo Moo** there
   Moo here **Moo** there, everywhere **Moo Moo**
   Old McDonald had a farm, E-I-E-I-O

   The lyrics can be generalized for different animals, each having a different *name* and **sound**. With knowledge of object-oriented programming, you want to demonstrate that it is possible to write a program that sings "Old McDonald". To show that your program works, add the following (*animal*, **sound**) to the list: (Cow, Moo), (Dog, Woof), and (Duck, Quack).

Use the following template to below to solve your problem. Don't forget to include the necessary file and namespace.

```
class Animal {
    /* TODO: Implement data and functionality of an Animal here */
};

class OldMcDonald {
private:
    vector<Animal*> _farm; // Old McDonald had a farm (still has now)

public:
    OldMcDonald() {
        /* TODO: Create your farm, an Animal* vector */
    }
    ~OldMcDonald() {
        /* TODO: Old McDonald has no (more) farm... */
    }

    void sing() {
        for (int i=0; i<_farm.size(); i++) {
            cout << "Old McDonald had a farm, E-I-E-I-O\n";
            /* TODO: Add the rest of the lyrics here... */
        }
    }
};
```