Exercise 4 of Programming Language II (CSE, NTOU)

ID: _____

11:10 – 12:00, 29 May 2026; Room INS101

Name: _____

Note: Cell phones and any smart device or calculator are forbidden.

1. (10%) Will the following code compile? Explain why `enum class` is safer than the old-style `enum`.

```
#include <iostream>

enum class Color {
    RED,
    GREEN,
    BLUE
};

enum class Fruit {
    APPLE,
    BANANA
};

int main() {
    Color c = Color::RED;
    Fruit f = Fruit::APPLE;

    if (c == f)
        std::cout << "same" << std::endl;
    else
        std::cout << "different" << std::endl;

    return 0;
}
```

2. (10%) Complete the following template function so that it swaps two variables of the same type.

```
template <class T>
void mySwap(__________, __________) {
    T temp = x;
    x = y;
    y = temp;
}
```

The following `main()` should work correctly:

```
int main() {
    int a = 5, b = 3;
    double c = 1.5, d = 9.2;

    mySwap(a, b);
    mySwap(c, d);

    cout << a << " " << b << endl;
    cout << c << " " << d << endl;

    return 0;
}
```

3. (10%) Consider the following code:

```
vector<int> v = {9, 7};

cout << v[1] << endl;     // line 1
cout << v.at(1) << endl; // line 2
cout << v[2] << endl;     // line 3
cout << v.at(2) << endl; // line 4
```

Which lines are safe? Which lines may cause problems? Explain the difference between `v[i]` and `v.at(i)`.

4. (10%) Given the following data:

```
vector<int> v = {10, 20, 30, 40, 50};
```

Write one statement to create a new vector **`sub`** containing only:

```
20, 30, 40
```

5. (10%) Recall that a **`deque`** supports insertion and deletion at both ends. Given:

```
deque<int> dq = {3, 4};
```

Write statements, using `push_front`, `push_back`, `pop_front`, and `pop_back`. so that `dq` becomes

```
1 2 3
```

6. (10%) Complete the following functor `Multiplier`. So that the main function prints `30`.

```
struct Multiplier {
    int factor;

    Multiplier(int f) : factor(f) {}

    int operator()(int x) const {
        // complete this line
    }
};

int main() {
    Multiplier times3(3);
    cout << times3(10) << endl;
    return 0;
}
```

7. (10%) Given `vector<int> v = { 2, 4, 6, 8 };` Complete the following code **using an iterator.**

```
for (_____________________________) {
    cout << *it << " ";
}
```

8. (10%) Complete the following C++11 program so that it rolls a fair six-sided die 20 times.

```
#include <iostream>
#include <iomanip>
#include <random>
#include <ctime>
using namespace std;

int main() {
    default_random_engine engine(_____________________________);
    uniform_int_distribution<unsigned int> randomInt(____, ____);

    for (unsigned int counter = 1; counter <= 20; ++counter) {
        cout << setw(10) << randomInt(engine);
        if (counter % 5 == 0)
            cout << endl;
    }

    return 0;
}
```

9. (10%) Complete the loop so that it prints each **key-value pair** in the **map**.

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

int main() {
    map<int, char> m;
    m[1] = 'A';
    m[3] = 'C';
    m[2] = 'B';

    for (const auto& __________ : m) {
        cout << _____._____ << " -> " << _____._____ << endl;
    }

    return 0;
}
```

Expected output:

```
1 -> A
2 -> B
3 -> C
```

10. (5%) Which STL container is usually used to support hash?

11. (10%) Complete the blanks in the following program.

```
#include <iostream>
#include <queue>
#include <vector>
#include <string>

struct Student {
    std::string name;
    double math;
    double science;
    double english;

    // Return the sum of the three scores.
    double total() const {
        return math + science + english;
    }
    // Define "operator<" so that the student with the smaller total
    // score has higher priority in the priority_queue.
    bool _______________(const Student &other) const {
        _____________________________________________;
    }
};
```

```
in main() {
    std::vector<Student> students = {
        {"Alice",   85.5, 92.0, 78.0},
        {"Bob",     90.0, 88.5, 91.0},
        {"Charlie", 70.0, 75.0, 80.0},
        {"Diana",   95.0, 85.0, 89.5},
        {"Evan",    88.0, 90.5, 92.0}
    };
    // Use the default comparator which will call
    // an overloaded operator internally.
    std::_____________________ pq; // fill in the blank
    for (const auto &s : students) {
        _________________ // please complete this line of inserting
                          // students in the priority queue pq
    }

    std::cout << "Worst 3 students by total score:\n";
    for (int rank = 1; rank <= 3 && !pq.empty(); ++rank) {
        const auto &s = __________; // complete the blank here.
        std::cout
            << rank << ". " << s.name
            << " (Total: " << s.total() << ")\n";
        pq.pop();
    }
    return 0;
}
```