CITY
UNIVERSITY OF LONDON
EST 1894

The University for
business
and the professions

School of Mathematics, Computer Science & Engineering

BSc Computer Science
BSc Computer Science with Computer Games Technology
MSci Computer Science
MSci Computer Science with Cyber Security
MSci Computer Science with Games Technology
MSci Data Science

IN2029: Programming in C++

Examination, January 2022

Answer TWO questions ONLY.
If more than TWO questions are answered, the best TWO will be counted.
Division of marks: All questions carry equal marks
BEGIN EACH QUESTION ON A FRESH PAGE

Number of answer books to be provided: ONE
Calculators permitted: Casio FX-83/85 MS/ES/GT+ ONLY
Examination duration: 90 minutes
Dictionaries permitted: English translation and language dictionaries are permitted
Additional materials: None
Can question paper be removed from the examination room: No

Question 1

a) Write a function that takes a list of `doubles` and returns the sum of the list. [25 Marks]

b) Write a function that takes a list of `doubles`, and modifies the list by replacing each 0 with 1. [25 Marks]

c) Using the following library algorithms:

count(b, e, v) returns the number of elements in the range $[b, e)$ that are equal to v.

find_if(b, e, p) searches the range $[b, e)$ and returns an iterator pointing at the first value for which p is `true`, if there is one, or e if there is not.

i) Write a function that takes a list of `ints` and an `int`, and returns the number of occurrences of that `int` in the list. [20 Marks]

ii) Write a function that takes a list of `ints` and returns the number of 0s between the first positive number and second positive number in the list. If the list contains no positive numbers, the function should return 0. If the list contains only one positive number, the function should return the number of 0s in the list after that. [30 Marks]

Question 2

a) Define a class `song`, with

  i) a constructor taking the title of the song, the name of the singer and its length in seconds.

  ii) accessor functions returning the title, singer and length.

[25 Marks]

b) Define a `playlist` class representing a collection of songs. Your class should include member functions to

  i) add a new song.

  ii) return the number of songs.

  iii) give the total length of all the songs in the playlist.

  iv) using the following library algorithm:

  count_if(b, e, p) returns the number of elements in the range $[b, e)$ for which p is `true`.

  return the number of songs by a given singer.

[75 Marks]

Question 3

Consider the class

```
class event {
    const string desc;

public:
    event(const string &d) : desc(d) {}

    const string & description() const { return desc; }

    // returns true if the event falls on the given date
    virtual bool on_day(int day, int month,
                        int year) const = 0;
};
```

a) Why can't we declare a variable of type `event`? [5 Marks]

b) Write a derived class `daily_event` of `event` describing an event that falls on every day. [30 Marks]

c) Write a derived class `annual_event` of `event` describing an event that falls on a particular day and month of each year. The day and month should be specified when the object is created. [40 Marks]

d) Show how you would represent a collection of various kinds of `event`, and how you would print the descriptions of all events in the collection that fall on a given day. [25 Marks]

Marking Scheme

Question 1

a) 25 marks

```
double sum_list(const list<double> &l) {
    double total = 0;
    for (auto x : l)
        total += x;
    return total;
}
```

signature [10, including 1 each for const and ref], loop structure [5], loop body [5], return [5]. A version using explicit `const_iterator`s is equally acceptable. 10 for function on vectors.

b) 25 marks

```
void replace_zeroes(list<double> &l) {
    for (auto &x : l)
        if (x == 0)
            x = 1;
}
```

Marking: signature [10, including 5 for the reference], loop structure [10, including 5 for the reference], loop body [5].

c) i) 20 marks

```
int occurrences(const list<int> &l, int n) {
    return count(l.cbegin(), l.cend(), n);
}
```

Marking: signature [10, including 2 for const ref], body [10].

ii) This can be done with an external function [7] 30 marks

```
bool positive(int n) {
    return n < 0;
}
```

Then our function is

```
int count_zeroes(const list<int> &l) {
    auto start = find_if(l.cbegin(), l.cend(), positive);
    if (start == l.cend())
        return 0;
    const auto finish = find_if(++start, l.cend(), positive);
    return count(start, finish, 0);
}
```

Alternatively, we could use a local lambda expression

```
    const auto positive = [] (int n) { return n < 0; };
```

Marking: signature [5], `find_if`s [10], ++ [2], `count` [6].

Question 2

a) 25 marks

```
class song {
    const string _title;
    const string _singer;
    const int _length;

public:
    song(const string &t, const string &s, int y) :
        _title(t), _singer(s), _length(y) {}

    const string & title() const { return _title; }
    const string & singer() const { return _singer; }
    int length() const { return _length; }
};
```

Marking: private [2] data members [3] (const optional), constructor [10 including 2 for const and ref], accessor functions [10, including 2 for consts and 1 for refs].

b) 75 marks

```
class playlist {
    vector<song> songs;

public:
    void add_song(const song &b) { songs.push_back(b); }

    int num_songs() const {
        return songs.size();
    }

    int total_length() const {
        int total = 0;
        for (const song &s : songs)
            total += s.length();
        return total;
    }

    int count_singer(const string & name) const {
        return count_if(songs.cbegin(), songs.cend(),
            [name &] (const song &s)
                { return s.singer() == name; });
    }
};
```

Member function definitions outside the class are equally acceptable.

In `total_length`, an explicit loop is equally acceptable:

```
        for (auto p = songs.cbegin(); p != songs.cend(); ++p)
            total += p->length();
```

Marking:

- private[1] vector (or other sequential container) [4],
- `add_song` [10, of which 2 for const ref],
- `num_songs` [10, of which 2 for const ref, 5 for body],
- `total_length` 20, divided as: signature [4, including 1 for const], loop [10, including 2 for const and not copying], init and return [6],
- `count_singer` 30, divided as: signature [5, including 3 for const ref and const], use of `count_if` [10], lambda expression [10, including 1 for capture by ref], test [5].

Question 3

a) The class is abstract, because it contains the pure virtual method on_day. 5 marks

b) 30 marks

```
class daily_event : public event {
public:
    daily_event(const string &d) : event(d) {}

    bool on_day(int day, int month, int year) const {
        return true;
    }
};
```

Marking: inheritance [5], constructor [15], overridden method [10].

c) 40 marks

```
class annual_event : public event {
    int _day, _month;

public:
    annual_event(const string &d, int dd, int mm) :
        event(d), _day(dd), _month(mm) {}

    bool on_day(int day, int month, int year) const {
        return day == _day && month == _month;
    }
};
```

Marking: inheritance [5], data members [5], constructor [15], overridden method [15].

d) Representing the collection [10]: 25 marks

```
vector<shared_ptr<event>> v;
```

(5 if `vector<event>`) Other sequential containers are also fine.

Printing matching descriptions:

```
for (const auto &ep : v)
    if (ep->on_day(d, m, y))
        cout << ep->desc() << '\n';
```

An explicit iterator version is also acceptable:

```
for (auto p = v.cbegin(); p != v.cend(); ++p)
    if ((*p)->on_day(d, m, y))
        cout << (*p)->desc() << '\n';
```

Marking: loop [5], test [5], printing [5].