Student Number _____

Student Name _____

School of Science and Engineering

FINAL EXAMINATION

Term 2, 2020-2021

CSC3002 Introduction to Computer Science:

Programming Paradigms

Examination Duration: 150 minutes.

This examination has 32 questions divided into 4 sections.

Exam Conditions:

This is a FORMAL Examination.

This is a RESTRICTED OPEN BOOK Examination.

Materials Permitted In The Exam Venue:

Maximum of one (1) sheet of double-sided A4 paper notes are permitted. NO OTHER MATERIALS ARE PERMITTED

Calculators are NOT ALLOWED in this examination.

2B pencils and erasers should be brought by the students and used to mark the answer sheet.

Materials To Be Supplied To Students:

This question and answer book, the answer sheet, and scratch paper.

Materials To Be Collected From Students:

EVERYTHING MUST BE RETURNED, including this question and answer book, the answer sheet, and scratch paper. Please answer the questions in the first 3 sections by marking the answer sheet, and those in the last section by writing down your answers in the specified places in this question and answer book.

Section 1. True or false questions. (10 × 1% = 10%)
Pick "a" for "True" or "b" for "False" for each of the following statements. Please mark your answers (only "a" or "b") on the answer sheet.

1. In C++, a **variable** is a named address for storing a type of value, and you must declare a variable before you can use it.

   a. True b. False

2. In C++, a **function** is a named section of code that performs a specific operation, and a function can include only one **`return`** statement as the exit point of the function.

   a. True b. False

3. The following two statements will generate two strings with the same content:

```
std::string str1 = "hello,world";
std::string str2 = "hello" + "," + std::string("world");
```

   a. True b. False

4. The insertion operator **<<** is often overloaded in different classes to insert the contents of the classes into an output stream for display purposes. The overloading usually requires the use of **return by reference**, because it avoids making a copy of the output stream. If you use **return by value**, a new copy of the output stream will be returned.

   a. True b. False

5. In a collection class, the **effective size** can never be equal to or greater than the **allocated size**.

   a. True b. False

6. The following statements declare two pointers **`p1`** and **`p3`**, and two integers **`p2`** and **`p4`**:

```
int*  p1, p2;
int  *p3, p4;
```

   a. True b. False

7. For a variable **`x`**, the expression **`*&x`** is essentially equivalent to **`x`**; and for a pointer **`p`**, the expression **`&*p`** is equivalent to **`p`**.

   a. True b. False

8. A **hash function** should distribute keys as uniformly as possible across the integer range, to minimize **collisions** and achieve $O(1)$ performance on **`put`** and **`get`** operations. If heavy collisions happen, the complexity of **`put`** and **`get`** will increase up to $O(\log N)$.

   a. True b. False

9. If **`c`** is a nonempty collection in C++ STL, calling **`c.begin()`** and **`c.end()`** will return **iterators** pointing at the first and last element of that collection, respectively.

   a. True b. False

10. C++ is a **multi-paradigm** programming language. Besides the **procedural** programming paradigm originally supported by C, C++ also supports the **object-oriented** paradigm. However, C++ does not support other paradigms such as the **functional** paradigm without external libraries.

    a. True b. False

Section 2. Single choice questions. (10 × 2% = 20%)
Pick the correct option in each of the following questions. Note that only ONE option is correct. Please mark your answers on the answer sheet.

11. Calculate the result of the following expressions based on the C++ rules:

```
9 * (8-7)/6 + 5*4 % 3*(2+1.0)
```

    a. `3`
    b. `3.5`
    c. `7`
    d. `7.5`

12. In the definition of a class, it is preferred that the data members be declared as:

    a. public
    b. protected
    c. private
    d. friend

13. Suppose that you are using the **Merge sort** algorithm to sort a vector of 1,000 integers and find that it takes T milliseconds to complete the operation. How many milliseconds would you expect the running time to be if you use the same algorithm to sort a vector of 1,000,000 integers on the same machine? (Assuming there is enough memory and $\log_2 1000 \approx 10$.)

    a. 1,000 T
    b. 2,000 T
    c. 10,000 T
    d. 1,000,000 T

14. If we have built a **36**-bit computer on which the memory address will be represented in one 36-bit long **machine word**, and the **smallest addressable unit** is a sequence of 8 consecutive bits called a byte, how much memory can this computer have, theoretically? (Assume 1KB = 2^{10} bytes, 1MB = 2^{10} KB, 1GB = 2^{10} MB.)

    a. 36 GB
    b. 64 GB
    c. 256 GB
    d. 2^{36} GB

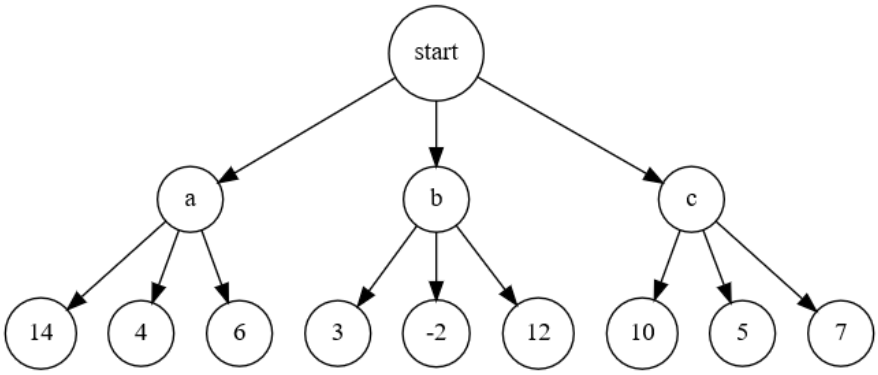
15. Assuming the following C++ variable declaration statement

```
int i = 2021, *pi = &i, **ppi = &pi;
```

    leads to the addresses of `i` and `pi` being `0x2020` and `0x2024`, respectively, what are the values of `ppi` and `**ppi`?

a. **`0x2020`** and **`2021`**
b. **`0x2024`** and **`2021`**
c. **`0x2020`** and **`0x2024`**
d. **`0x2024`** and **`0x2028`**

16. In a two-player game, suppose you are in a position in which the analysis for the next two moves shows the following rated outcomes (larger values represent better outcomes for you) from your original player's point-of-view:



If you adopt the **minimax** strategy in your program, what is the best move and the rating of that move from your perspective?

a. **`a, 14`**
b. **`b, 12`**
c. **`c, 5`**
d. **`c, 10`**

17. Computers typically represent each pixel in an image as a 32-bit integer, in which the 4 bytes are interpreted as the α (Transparency), Red, Green, and Blue components. The following bitwise operations extracts one of the components from the hexadecimal representation of the αRGB color stored in an integer variable **`pixel`**. Which color is extracted and what is the component value in **decimal** format?

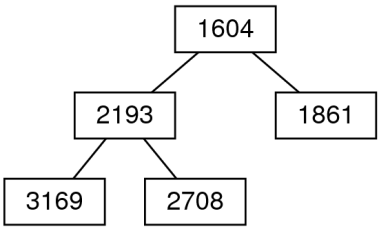
```
int pixel = 0xFF202104;
int color = (pixel >> 8) & 0xFF;
```

a. α, 0
b. α, 255
c. Green, 21
d. Green, 33

18. If we build a binary search tree by inserting the nodes in increasing order (i.e., the smaller the key is, the earlier the node is inserted), and do not balance it, the result will be an extremely unbalanced tree, i.e., a linked list. If we then traverse this tree and **`cout`** the key, which one of the following traversal approaches will have different output from the others?

a. Preorder
b. Inorder
c. Postorder
d. Level-order

19. Suppose that you are modelling a **priority queue** with a **partially ordered tree** that contains the following data (assuming smaller values have higher priorities):



What is the proper content in the corresponding **heap** after deleting a node and then inserting 2021?

a. 1604, 2021, 1861, 3169, 2193
b. 1861, 2021, 2708, 3169, 2193
c. 1861, 2021, 2193, 2708, 3169
d. 1861, 2193, 2021, 3169, 2708

20. If the parameter you are passing is an object whose value should not change, and you don't want to make a copy of the object because it occupies a lot of memory, which one of the following approaches should you use to pass this object?

a. Call by value
b. Call by reference
c. Call by pointer
d. Constant call by reference

Section 3. Multiple choice questions. (10 × 4% = 40%)
Pick the correct option(s) in each of the following questions. Note that there may be ONE to FOUR correct options for each question. Incomplete answers will get partial scores (2%). Please mark your answers on the answer sheet.

21. What are the **distinguishing** features of **object-oriented** programming languages like C++ (i.e., what makes object-oriented programming different from other programming paradigms)?

a. Abstraction
b. Encapsulation
c. Inheritance
d. Polymorphism

22. In C++, functions can be **overloaded**, which means that you can define different functions with the same name as long as they have different **signatures**. The signature of a function includes the following:

a. The number of the parameters
b. The names of the parameters
c. The types of the parameters
d. The type of the return value

23. Which of the following collection classes in the Stanford C++ Library do not support the use of the range-based `for` loop?

a. **`Vector`**
b. **`Stack`**
c. **`Queue`**
d. **`Map`**

24. Indicate which of the following are **illegal** variable names in C++:

a. **`typename`**
b. **`type_name`**
c. **`type-name`**
d. **`2typenames`**

25. Which of the following are the areas of memory in which values can be stored in a C++ program?

a. Static area
b. Heap
c. Stack
d. Queue

26. Normally in C++ you should not use pointers as much as in the C language, in which pointers are important for the following reasons:

a. Pointers make it possible to reserve new memory during program execution.
b. Pointers allow you to specify the memory addresses for dynamically allocated variables and objects in the heap.
c. Pointers allow you to refer to a large data structure in a compact way.
d. Pointers can be used to record relationships among data items, e.g., linked list.

27. In C++, any expression that refers to an internal memory location capable of storing data is called an **lvalue**, which can appear on the left side of an assignment statement. Which of the following terms in a C++ program **cannot** be lvalues?

a. **`p`**
b. **`*p`**
c. **`&p`**
d. **`(p+1)`**

28. What **properties** must a problem have for **recursion** to make sense as a solution strategy?

a. There must be simple cases or exit points for which the answer is easily determined, without using recursion.
b. There must be a function that calls itself.
c. There must be a recursive decomposition that allows you to break any complex instance of the problem into simpler problems of the same form.
d. There must be an inclusion-exclusion pattern in the problem.

29. A **ring buffer** is an array whose end is linked back to the beginning. In a typical ring buffer based queue implementation as follows,

```
ValueType *array;
int capacity;
int head;
int tail;
```

the **head** field holds the index of the next element to come out of the queue, and the **tail** field holds the index of the next free slot. If we use the equality of the **head** and **tail** fields to indicate an empty queue, which of the following cases may indicate that the queue is full?

a. `head == tail`
b. `head - tail == 1`
c. `tail - head == capacity`
d. `tail - head == capacity - 1`

30. After running the following program, which of the numbers will be seen in the output?

```
#include <iostream>
using namespace std;
class A {
public:
    int a;
    A() { a = 1; }
    void display() { cout << a << endl; }
};
class B: public A {
public:
    int b;
    B() { b = 2; }
    void display() { cout << a << b << endl; }
};
class C: public B {
public:
    int c;
    C() { b = 1; c = 3; }
    virtual void display() { cout << a << b << c << endl; }
};
int main() {
    C oC;
    B* pb = &oC;
    pb->display();
    return 0;
}
```

a. `1`
b. `2`
c. `3`
d. None, compile error.

Section 4. Fill-in-the-Blank Questions. (12%+18% = 30%)
Answer the following questions. Please write your answers on THIS EXAM PAPER in the blank space after the questions.

31. What is the formatted output of the following program? Write your answer in the box, and make sure the format is correct. (12%)

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

```
int main(void)
{
    int arr[] = {2, 0, 2, 1};
    int* p = arr;
    int a = *++p;
    int b = *p++;
    int c = ++*p;
    int d = (*p)++;
    cout << arr[0] << arr[1] << arr[2] << arr[3] << endl;
    cout << a << b << c << d << endl;
    return 0;
}
```

32. Assume that **doubleArray** is declared as

```
double doubleArray[] = {0, 1, 1, 2, 3, 5, 8, 13, 21, 34};
```

and variables of type double take up **8 bytes** on the computer system you are using. If the base address of the array **doubleArray** happens to be **0xACE0**, what are the values of the following 12 expressions? Write your answer in the boxes below, **all in 4-character hexadecimal format**, e.g., 0 should be 0000. (12 x 1.5% = 18%)

doubleArray	= 0x	ACE0
*doubleArray	= 0x	
&doubleArray	= 0x	
&doubleArray[0]	= 0x	
doubleArray[0]	= 0x	
doubleArray+1	= 0x	
&doubleArray+1	= 0x	
&doubleArray[1]	= 0x	
doubleArray+9	= 0x	
*doubleArray+9	= 0x	
*(doubleArray+9)	= 0x	
&doubleArray[9]	= 0x	
doubleArray[9]	= 0x	

END OF EXAMINATION