FULL NAME Daniel Ben Gifen

STUDENT ID 300447549

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI

VICTORIA
UNIVERSITY OF WELLINGTON

A	6
B	10
C	23
	39

MID-TERM TEST – 2019

TRIMESTER 1

NWEN241

SYSTEMS PROGRAMMING

Time allowed: 45 MINUTES

CLOSED BOOK

Permitted materials: Only silent non-programmable calculators or silent programmable calculators with their memories cleared are permitted in this examination.
No electronic dictionaries are allowed.
Paper foreign to English language dictionaries are allowed.

Instructions: Attempt ALL TWENTY-FOUR (24) questions.

There are THREE sections:

- SECTION A – True or False [8 marks]
- SECTION B – MCQ [12 marks]
- SECTION C – Short Answer [25 marks]

The examination consists of 45 marks in total.

You must use the answer sheet provided for Sections A (Questions 1-8) and B (Questions 9-20). For Section C (Questions 21-24), you must write your answers in the boxes provided within this questionnaire.

Submit both the answer sheet and this questionnaire.

Student ID: ....

SECTION A True or False

Use the answer sheet provided for answering the questions in this section. Each correct answer will garner 1 mark.

1. An abstract class is a class that has at most one pure virtual function member.

   (a) True

   (b) False

2. One key difference between C++ structures and C structures is that C++ structures can have member functions.

   (a) True

   (b) False

3. In the assembly phase, the compiler translates a pre-processed C/C++ source code into an assembly file.

   (a) True

   (b) False

4. In C and C++, there is no check on whether an array index is out of bounds. If an array index goes out of bounds, the program *always* terminates in an error.

   (a) True

   (b) False

5. Arrays can be initialised during their declaration. If there are fewer initial values than the array size, the remaining elements are initialized to 0.

   (a) True

   (b) False

6. The base address of an array is the address of the first array component.

   (a) True

   (b) False

7. The following C code will compile without errors:

   ```
   int foo(const int * a, const int * b)
   {
       *a = 5;
       return *a + *b;
   }
   ```

   (a) True

   (b) False

8. In C++, the statement `delete p;` deallocates the variable pointer `p`.

   (a) True

   (b) False

Student ID: ....

SECTION B Multiple Choice

Use the answer sheet provided for answering the questions in this section. Each correct answer will garner 1 mark.

9. Consider the following statement:

   ```
   char str[] = "Twelve";
   ```

   What is the size of the array `str`?

   (a) The statement will cause a syntax error.

   (b) 6

   (c) 7

   (d) None of the above

10. Consider the following code fragment:

    ```
    int i = 4, j = 0;
    while(--i) { j++; }
    ```

    What is the value of the variable `j` after the completion of the while-loop?

    (a) The code will not compile because of syntax error in the while-loop condition.

    (b) 2

    (c) 3

    (d) 4

11. Consider the following code snippet:

    ```
    union {
        char c;
        short s;
        int i;
    } u;

    u.c = 'A';
    ```

    What is the size of the variable `u` equal to?

    (a) `sizeof(char)`

    (b) `sizeof(short)`

    (c) `sizeof(int)`

    (d) None of the above

Student ID: ....

12. Consider the following function-like macro:

    `#define FLM(X,Y) X/Y`

    To what value does the macro evaluate to when invoked as `FLM(2 + 9, 3 - 2)`?

    (Hint: The operator / has higher precedence than + and –. The operators + and – have the same precedence, with left to right associativity.)

    (a) 11

    (b) 3

    (c) 1

    (d) None of the above

13. Consider the following C/C++ code snippet:

```
                                 0              2               3
enum loudness { faint = -1, moderate, defeaning = 2, painful };
```

    What is the value of `moderate`?

    (a) 0

    (b) 1

    (c) 2

    (d) None of the above

14. In C and C++, generic pointers can be declared with

    (a) `static`

    (b) `void`

    (c) `const`

    (d) None of the above

15. Given the declaration below:

    `char name[30];`

    Which of the following statements are *invalid*?

    ✓ i. `name = "Terakihi";`

    ✓ ii. `strcpy(name, "Snapper");`

    ✓ iii. `name = {'H', 'a', 'k', 'e'};`

    ✓ iv. `name[0] = 'G';`

    (a) i and iii

    (b) ii and iv

    (c) i and ii

    (d) They are all valid

16. What is the output from the following C++ code?

```
#include <iostream>
using namespace std;
int main(void)
{
    int * ptr = new int;
    cout<<ptr<<" - "<<*ptr;
    return 0;
}
```

(a) (Address of memory allocated) - (Garbage value)

(b) (Address of memory allocated) - 0

(c) (Address of `ptr`) - (Garbage value)

(d) (Address of `ptr`) - 0

17. Given the C++ declaration:

`char name[8] = "Marlin";`

Which of the following statements output `Marlin`?

i. `std::cout << name;`

ii. `for(int j=0; j<6; j++) std::cout << name[j];`

iii. `int j=0; while (name[j] != '\0') std::cout << name[j++];`

iv. `int j=0; while (j < 8) std::cout << name[j+1];`

(a) All of the above

(b) None of the above

(c) i, ii and iii

(d) i, iii and iv

18. What is the name of the `&` operator in relation to pointers?

(a) Conditional operator

(b) Logical operator

(c) Address of operator

(d) None of the above

19. What is the output of the following program?

```
#include<stdio.h>
int main()
{
    int a[5] = {2, 3};
    printf("%d,%d,%d\n", a[2], a[3], a[4]);
    return 0;
}
```

(a) Garbage values

(b) 2,3,3

(c) 3,2,2

(d) 0,0,0

20. What would be the equivalent pointer expression for referring to the array element `a[i][j][k][l]`?

    (a) `((((a+i)+j)+k)+l)`

    (b) `*(*(*(*(a+i)+j)+k)+l)`

    (c) `(((a+i)+j)+k+l)`

    (d) `((a+i)+j+k+l)`

pcall

Student ID: ....

SECTION C Short Answer

Write your answer in the space provided.

21. What are the four phases of compilation in C and C++? **(8 marks)**

~~preprossesing~~ pre prosessing
compiling.
assembly
linking.

✓ 8

22. Consider the following C++ class declaration: **(5 marks)**

```
namespace nsA {
    class ClassA {
    public:
        virtual int f1() const = 0;
        virtual void f2() = 0;
    protected:
        int a;
    };
}
```

4

Declare a class `ClassB` that extends `ClassA` but in a different namespace called `nsB`. `ClassB` should preserve the access specifier of the members, should not be abstract, and should have an inline default constructor that initializes the member variable `a` to `100`.

(Hint: You do not need to show function implementations, just the prototype declarations)

~~using namespace nsA~~
~~class ClassB : ClassA pu'~~
~~public~~

public

```
namespace nsB {
    class ClassB : protected nsA::ClassA {
        public:
            int f1() const; ✓
            void f2() ; ✓
            ClassB() {nsA::ClassA.a =100; } ✓
    };
}
```

23. Given these two variable declarations in a C program:

6

```
char str1[] = "I am a string.\n";
char *str2 = "I am a string.\n";
```

(a) What is the difference between the two statements? **(1 mark)**

They differ only in notation, ~~but~~
~~but~~

X

0

(b) Write a statement to output the letter ”s” in `str1` using `printf`. Use array index to refer to the element. **(2 marks)**

```
printf("%c", str1[7]);
```

✓ 2

(c) Write a statement to output the letter ”s” in `str2` using `printf`. Use appropriate pointer arithmetic and operator to refer to the element. **(2 marks)**

```
printf("%c", *(str2+7));
```

✓ 2

(d) Write some code to copy the second string to the first. Which header file do you need to include? **(2 marks)**

(Hint: The function `char *strcpy(char *dest, const char *src)` copies the string pointed to by `src` to `dest`.)

Code to copy second string to first:

```
strcpy(str1, str2);
```

✓ 2

Header file to include:

```
#include <cstring.h>
```

✓

24. Given the following code:

```
char Z, Y;
Z ='g';
Y ='#';
char * pZ,* pY;
pY = &Y;
pZ = &Z;
pY = pZ;
```

5

Suppose each `char` occupies 1 byte of memory and the variable `Z` is at (decimal) address 2304 and the variable `Y` is at (decimal) address 2305. After the code above has run:

(a) What value is represented by `&Y`? **(1 mark)**

2305 ✓, the mem address of Y

(b) What value is represented by `pZ`? **(1 mark)**

2304, ✓ the mem address of Z

(c) What value is represented by `pY`? **(1 mark)**

2304, ✓ address of Z

(d) What value is represented by `*pZ`? **(1 mark)**

g ✓

(e) What is the value represented by `*pY`? **(1 mark)**

g ✓

* * * * * * * * * * * * * * *