



EXAMINATIONS – 2023

TRIMESTER 1

<p>NWEN 241 Systems Programming 21 June 2023</p>
--

Time Allowed: TWO HOURS

CLOSED BOOK (SELECTED MATERIALS ONLY)

Permitted materials: Only silent non-programmable calculators or silent programmable calculators with their memories cleared are permitted in this examination.

Paper foreign to English language dictionaries are allowed.

Electronic dictionaries are NOT allowed.

NO other material is allowed.

Instructions: Attempt all **SEVEN** questions.

The questions are grouped into two main sections:

- SECTION A: C Programming [40 Marks]
- SECTION B: Process Management and C++ Programming [80 marks]

All answers must be written in the boxes provided in this test paper.

The examination consists of **120** marks in total.

Within each question, the marks for subparts are shown.

SECTION A C programming (40 marks)

1. Arrays, Strings and Pointers

(15 marks)

- (a) Fill in the blanks (i) and (ii) such that the following C program would be able to traverse the entire array arr. (4 marks)

```
#include <stdio.h>

int main(void) {
    int arr[10] = {1 2,3,4,5};

    for (int * p = _____ /* (i) */ ; p < _____ /* (ii) */ ; p++)
        printf("%d ", * p);

    return 0;
}
```

Answer for (i):

Answer for (ii):

- (b) Consider the following C program:

(2 marks)

```
#include <stdio.h>

#define HELLO "Hello"
#define WORLD "World"

int main(void) {
    char msg[] = HELLO WORLD;
    printf("%s %d", msg, strlen(msg));
}
```

Will the program compile, and if so, what is its output?

- (c) Fill in the blanks (i)–(iv) such that the following function would return the number of occurrences of letter the 'a' (both upper and lower case) in the input str which is a null-terminated string. (You are **not** allowed to declare any other variable.)
(4 marks)

```
int count_As(char * str) {  
    int count = 0;  
    while (_____ /* (i) */ ) {  
        if ( * str == 'A' || * str == 'a')  
            _____; /* (ii) */  
            _____; /* (iii) */  
        }  
        _____; /* (iv) */  
    }  
}
```

Answer for (i):

Answer for (ii):

Answer for (iii):

Answer for (iv):

(d) Given the following variable declarations:

```
short a[] = {2, 4, 8, 16, 32, 64};  
short *p = a;  
short **pp = &p;
```

Suppose that a short occupies 2 bytes in memory. The array a is at memory address 100, while p is at memory address 200 (all addresses are in decimal).

i. What is the numeric value of the expression p? **(1 mark)**

ii. What is the numeric value of the expression *p+1? **(1 mark)**

iii. True or False: &a[2] == p+2 **(1 mark)**

iv. What is the numeric value of the expression pp? **(1 mark)**

v. Using the pointer to pointer pp, write an expression that accesses the second element of array a, that is, a[1]. **(1 mark)**

2. Structures & Dynamic Memory

(10 marks)

- (a) Consider the following C structure, with tag `point3d` and consisting of 3 float members `x`, `y` and `z`: (2 marks)

```
struct point3d {  
    float x;  
    float y;  
    float z;  
};
```

Use typedef to define a new type `point3d_t` from the above structure.

- (b) Consider the following declaration where `struct point3d` is the structure defined in (a): (2 marks)

```
struct point3d *p;
```

Write a C statement using `calloc()` that will dynamically allocate an array of 15 `struct point3d` elements, and let `p` point to that memory.

(c) Implement the function with prototype

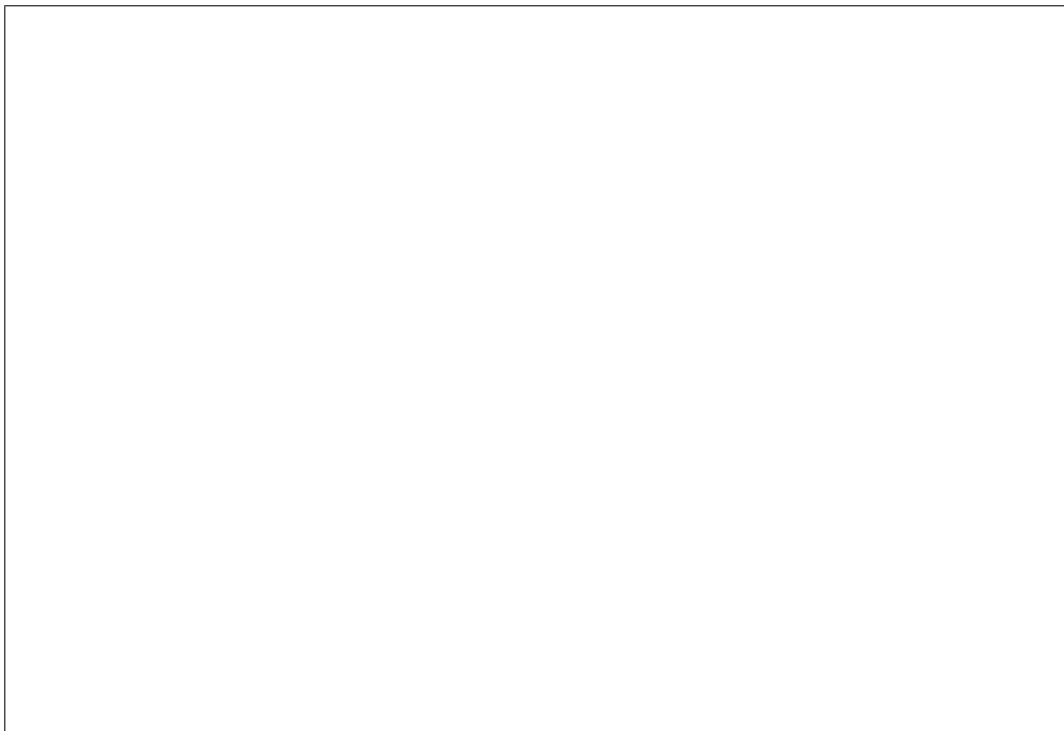
(4 marks)

```
struct point3d *new_point3d(float x_, float y_, float z_);
```

that will dynamically allocate memory for a `struct point3d *` using `malloc()`, initialize the members `x`, `y` and `z` to `x_`, `y_` and `z_`, respectively, and return a pointer to the allocated memory. If the memory allocation fails, the function should return `NULL`.



(d) Write a short C code snippet to illustrate the problem of memory leak. **(2 marks)**



3. File Stream I/O and Command Line Arguments

(15 marks)

Study the following C program which contains blanks.

```
#include <stdio.h>

#define INPUT_FILE "input.txt"

int main(int argc, char * argv[]) {
    FILE * infp, * outfp = NULL;
    int c, d;

    if (argc == 1)
        _____; /* question (a) */
    else if (argc == 2)
        _____; /* question (b) */

    if (infp == NULL)
        return 0;

    while (_____ /* question (c) */ ) {
        _____; /* question (d) */
        d = toupper(c);
        if (outfp != NULL)
            _____; /* question (e) */
    }

    if (outfp != NULL)
        fclose(outfp);
    fclose(infp);

    return 0;
}
```

- (a) Write a single C statement to open file INPUT_FILE for reading, and assign the opened file stream to infp. **(2 marks)**

- (b) Write a single C statement to open the file passed as the first command line argument for writing, and assign the opened file stream to `outfp`. **(2 marks)**

- (c) Write an expression using `feof()` that tests the end-of-file of `infp` is not yet reached. **(2 marks)**

- (d) Write a single C statement that reads a single character from the stream `infp` and stores it in `c`. **(2 marks)**

- (e) Write a single C statement that outputs `d` as a single character to the stream `outfp`. **(2 marks)**

- (f) Suppose that you have completed the program correctly and compiled the code to a binary executable file named `execfile`. Suppose further that you executed the file under a directory where a file named `input.txt` exists which contains

Hello world

- i. If the command you entered in the terminal was **(2 marks)**

`./execfile hello.txt`

what would be the contents of `hello.txt`?

ii. If the command you entered in the terminal was

(3 marks)

```
./execfile input.txt
```

what would be the contents of `input.txt` after execution? Briefly explain your answer.

SECTION B Process Management and C++ programming (80 marks)

4. Process Management

(15 marks)

(a) How many times will the following C program print NWEN241?

(3 marks)

```
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>

int main() {
    fork() && fork();
    fork();
    fork();
    printf("NWEN241\n");
}
```

(b) Name the process that owns the process ID 1 in Linux operating system. **(2 marks)**

(c) Fill in the blank. Fork system call returns _____ to the parent process on success. **(2 marks)**

(d) Which of the following system calls used in socket programming is a blocking system call? **(2 marks)**

- (a) accept
- (b) socket
- (c) bind
- (d) listen

(e) In which of the following system calls an identical copy of the original process is created? **(2 marks)**

- (a) execl
- (b) fork
- (c) accept
- (d) wait

(f) You are given the following C program.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <unistd.h>
4. #include <sys/wait.h>
5.
6. int gvar = 6;
7.
8. int main(void) {
9.     int lvar = 3;
10.    pid_t pid;
11.
12.    if((pid = fork()) < 0) {
13.        printf("fork error\n");
14.    }
15.    if(pid == 0) {
16.        gvar++;
17.        lvar++;
18.    } else {
19.        wait(NULL);
20.    }
21.    printf("%ld %d %d\n", (long) getpid(), gvar, lvar);
22.    exit(0);
23.}
```

i. Which part of the program code is relevant to the child process created by the fork system call at line number 12. **(2 marks)**

- ii. Assume that the fork is successful and that the parent process ID is 12345 while the child process ID is 12346. What is the output of the program? (2 marks)

5. C++ Classes

(30 marks)

- (a) Consider the following code segment.

(2 marks)

```
1. #include<iostream>
2.
3. class A {
4.     private: int item = 0;
5.     char icode = 'A';
6.     public:
7.         A():item(0),icode('A'){ }
8.         A(int x, char c):item(x),icode(c) { }
9.         A(A& a):item(a.item),icode(a.icode) { }
10.};
11.
12.int main() {
13.    A a1(1, 'A');
14.    A a2 = a1;
15.    std::cout << "Two objects created";
16.    return 0;
17.}
```

Which constructor type is invoked at line number 14?

- (a) Default constructor
- (b) Parameterized constructor
- (c) Copy constructor
- (d) No constructor is invoked

- (b) Consider the following code segment. In which order are the constructors invoked when an object `c` of class `C` is created? **(2 marks)**

```
class A {
    int a;
public:
    A() {
        std::cout << "A";
    }
};

class B {
    int b;
public:
    B() {
        std::cout << "B";
    }
};

class C: public B, public A {
    int c;
public:
    C() {
        std::cout << "C";
    }
};
```

- (c) State True or False. Declaring a constructor as `explicit` prevents implicit casting of constructor arguments to class objects. **(1 mark)**

- (d) State True or False. In C++ the statement `delete p;` deallocates the memory pointed-to by the variable `p` and not the variable `p`. **(1 mark)**

- (e) Consider the following code segment. Write the statement to access the variable a of the namespace Box1. **(2 marks)**

```
namespace Box1 {  
    int a = 4;  
}
```

```
namespace Box2 {  
    int a = 12;  
}
```

- (f) List one difference between a friend function and a member function of a class. **(2 marks)**

- (g) Given a class Student, which form of the operator delete would you use to deallocate the memory allocated by this statement: **(2 marks)**

```
Student *student_list = new Student[10];
```

- (a) delete student_list;
- (b) delete []student_list;
- (c) delete [10]student_list;
- (d) delete student_list[10];

(h) Consider the following code snippet.

```
#include<iostream>

class A {
public: int a;
void f1(void) {
    a = 10;
}

void f2(void) {
    aa = 20;
}

int f3(void) const {
    f1();
    f2();
    return 0;
}
protected: int aa;

private: int aaa;
};

int main() {
    A objA;
    objA.f3();
    return 0;
}
```

For an object objA of class A, state whether the following statements are True or False.

i. The statement `ObjA . aa = 10;` is valid **(2 marks)**

ii. The statement `ObjA . aaa = 10;` is valid. **(2 marks)**

iii. The statement `ObjA.a = 10;` is valid. **(2 marks)**

iv. It is not possible to instantiate `ObjA` as there is no constructor defined in the class definition. **(2 marks)**

v. Calling the member functions `f1()` and `f2()` from within the member function `f3()` is valid. **(2 marks)**

(i) What will be the output of the following program. **(2 marks)**

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

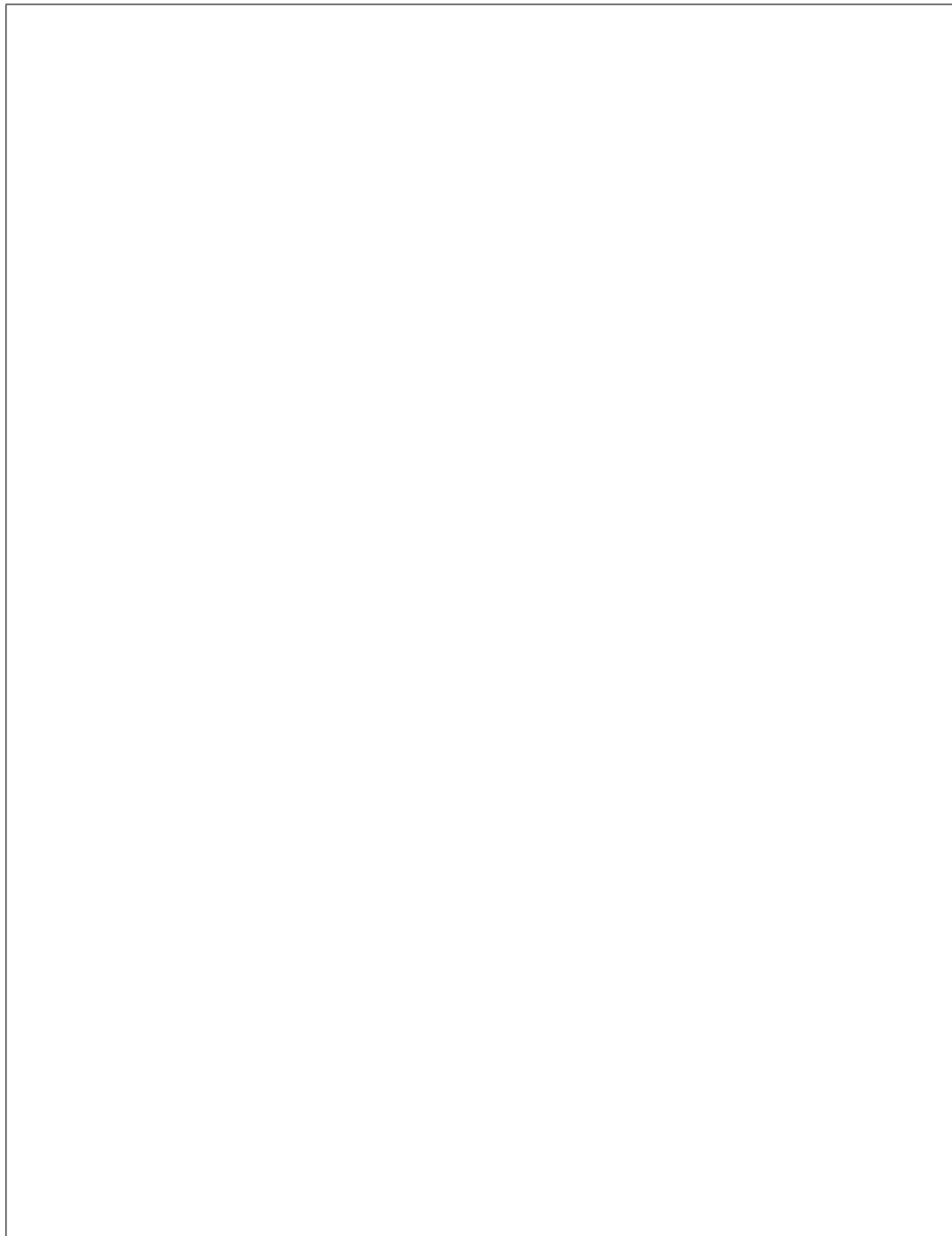
class Myclass {
    int x;
public:
    Myclass() {
        x = 5;
    }
};

int main() {
    Myclass * c = new Myclass;
    cout << c->x;
}
```


(j) Define a class Cube with the following members:

(4 marks)

- a private integer member `side`
- a public default constructor that initializes the data member `side` to 1
- a public parameterized constructor that takes in an integer to initialize `side` using initializer list
- a destructor that prints a "Destructor invoked" message when an object of Cube gets destroyed
- a public method `volume` that returns volume of the Cube object for which it is invoked.



(k) What is the difference between the following two statements.

(2 marks)

```
int* p = new int[5];  
int* p = new int(5);
```

6. Templates and Containers

(25 marks)

(a) What will be the output of the following C++ program.

(5 marks)

```
#include <iostream>  
using namespace std;  
  
template < typename T >  
void display(const T& x) {  
    static int count = 0;  
    cout << "x = " << x << " count = " << count << endl;  
    ++count;  
    return;  
}  
  
int main() {  
    display<int>(1);  
    display<int>(2);  
    display<double>(1.2);  
    return 0;  
}
```

(b) What will be the output of the following C++ program.

(4 marks)

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

template < typename T >
T max(T& p, T& q) {
    return (p > q ? p : q);
}

int main() {
    int x = 155, y = 60, m;
    long a = 105, b = 59, n;
    m = max(x, y);
    n = max(a, b);
    cout << m << endl;
    cout << n << endl;
    return 0;
}
```

(c) Name two functions used by iterators to support traversing of containers in C++.

(2 marks)

(d) There are two possible types of access that a container can support: linear and random. What type of access does the following two containers provide? **(3 marks)**

(a) vector

(b) list

(e) What will be the output of the following C++ program.

(4 marks)

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

void showlist(list<int> l) {
    list<int>::iterator it;
    for (it = l.begin(); it != l.end(); ++it)
        cout << * it << " ";
    cout << "\n";
}

int main() {
    list<int> list1;
    for (int i = 1; i < 10; i = i + 2) {
        list1.push_back(i);
    }
    list1.pop_front();
    list1.reverse();
    showlist(list1);
    return 0;
}
```

(f) What will be the output of the following program?

(4 marks)

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

int main() {
    vector<int> myvector;
    myvector.push_back(78);
    myvector.push_back(16);
    myvector.front() += myvector.back();
    cout << myvector.front() << '\n';
    return 0;
}
```

- (g) Give three advantages of using the generic vector class over a C-style array.
(3 marks)

7. File Handling

(10 marks)

- (a) In C++, which of the following will read an entire line from keyboard and store it in a `std::string` variable `str`? **(2 marks)**

- (a) `std::cin >> str;`
- (b) `std::cin << str;`
- (c) `std::getline(std::cin, str);`
- (d) `std::gets(std::cin, str);`

(b) Which of the following is a correct statement to read a maximum of 20 characters into a C-string s until the extracted character is 't'? **(2 marks)**

- (a) `std::cin.getline(s, 20, 't');`
- (b) `std::cin.getline(s, 21, 't');`
- (c) `std::cin.get(s, 20, 't');`
- (d) `std::cin.get(s, 21, 't');`

- (a) ii only
- (b) iii only
- (c) ii and iv only
- (d) i and iii only

(c) Give one difference between `std::get()` and `std::getline()` functions. **(2 marks)**

(d) Write a C++ code that will declare and open a text file `records.csv` for input. **(2 marks)**

(e) Which of the following is not used as a file opening mode? **(2 marks)**

- (a) `ios::trunc`
- (b) `ios::binary`
- (c) `ios::in`
- (d) `ios::ate`
