VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

EXAMINATIONS – 2025

TRIMESTER 1

NWEN 241

Systems Programming

16/06/2025

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 **FIVE** questions.

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.

1. **System Calls, Process Management and Socket Programming** **(25 marks)**

(a) What is the process ID (`PID`) of the `init` process on most Linux systems? **(2 marks)**

(i) 0

(ii) 1

(iii) 100

(iv) It varies depending on the system

(b) Which one of the following commands displays **all** processes running on the system, not just those associated with the current terminal? **(2 marks)**

(i) `ls`

(ii) `ps`

(iii) `ps -ef`

(iv) `ls -l`

(c) Which one of the following statements regarding the `exec()` family of system calls is TRUE? **(2 marks)**

(i) `exec()` creates a new process and allows the old process to continue running.

(ii) `exec()` replaces the current process image with a new program.

(iii) `exec()` cannot be called by a process that has previously called `fork()`.

(iv) `exec()` returns the process ID of the new program upon success.

(d) A 32-bit integer `0x12345678` is stored in memory starting at address `0x1000` on a little-endian system. What is the byte stored at address `0x1000`? **(2 marks)**

(i) `0x12`

ii) `0x34`

iii) `0x56`

iv) `0x78`

(e) Which one of the following correctly matches a process state with its meaning? **(2 marks)**

(i) New: The process is loaded in memory and ready to execute.

(ii) Running: The process is currently being executed by the CPU.

(iii) Waiting: The process is ready and waiting for CPU scheduling.

(iv) Terminated: The process is paused and waiting for an event to resume.

(f) Consider the following C code. Identify which function(s) may invoke a system call, either directly or indirectly. **(2 marks)**

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
void main() {
     printf("Starting program...\n");
     double result = cos(0.0);
     char *buffer = (char *)malloc(100);
     if (buffer != NULL) {
          printf("Memory allocated.\n");
          free(buffer);
       }
  }
```

(i) `printf()` only

(ii) `malloc()` only

(iii) Both `printf()` and `malloc()`

(iv) Both `printf()` and `cos()`

(g) State True or False. The accept() system call is blocking by default. **(2 marks)**

(h) Consider the following C program segment. What is the purpose of the `htons(8080)` function call? **(2 marks)**

```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>

   int main() {
     int sockfd = socket(AF_INET, SOCK_STREAM, 0);
     struct sockaddr_in addr;
     addr.sin_family = AF_INET;
     addr.sin_port = htons(8080);   //call to htons
     addr.sin_addr.s_addr = INADDR_ANY;

      if (bind(sockfd, (struct sockaddr *)&addr, sizeof(addr)) < 0) {
         perror("Bind failed");
         return 1;
     }

     printf("Bind successful\n");
     return 0;
}
```

(i) It converts the port number to the host's native byte order.

(ii) It ensures the port number is in network byte order (big-endian).

(iii) It checks whether port 8080 is available for binding.

(iv) It sets the socket to use the HTTP protocol on port 8080.

(i) Which socket type and protocol combination would you use to implement a reliable connection between a client and a server? **(2 marks)**

(i) `SOCK_DGRAM` with `UDP`

(ii) `SOCK_STREAM` with `TCP`

(iii) `SOCK_STREAM` with `UDP`

(iv) `SOCK_DGRAM` with `TCP`

(j) In the following C program, how many child processes are created in total when the code is executed? **(3 marks)**

```
#include <stdio.h>
#include <unistd.h>

int main() {
    fork();
    fork() && fork();
    printf("Hello\n");
    return 0;
}
```

(i) 4

(ii) 6

(iii) 5

(iv) 7

(k) Discuss briefly the two steps involved in establishing a socket in a connection-oriented client process, stating the specific system call invoked in the step. Do not include the actual sending and receiving of data in the steps. **(4 marks)**

2. **C++ Basics** **(25 marks)**

(a) State True or False. `std::cerr` is buffered by default. **(2 marks)**

(b) Which one of the following statements is TRUE about `wchar_t` in C++? **(2 marks)**

(i) `wchar_t` is guaranteed to be 2 bytes on all platforms.

(ii) `wchar_t` is always encoded in UTF-16.

(iii) The size of `wchar_t` is implementation-defined and can vary between platforms.

(iv) `wchar_t` can store multiple characters in a single variable.

(c) What will be the output of the following C++ program? **(2 marks)**

```
#include <iostream>

class Test {
public:
    Test() {
        std::cout << "Default Constructor called" << std::endl;
    }

    Test(const Test &t) {
        std::cout << "Copy constructor called" << std::endl;
    }
};

int main() {
    Test t1;
    Test t2 = t1;
}
```

(d) What will be the output of the following C++ program? **(2 marks)**

```
#include <iostream>
namespace A {
    void display() {
        std::cout << "Namespace A display function" << std::endl;
   }
    class Printer {
    public:
        void display() {
            std::cout << "Printer class display method" << std::endl;
        }
    };
}
int main() {
    using namespace A;

    display();  // Line 1

    Printer p;
    p.display();  // Line 2

    return 0;
}
```

(i) It calls the `display()` function from `namespace A` and then calls the `display()` method of the `Printer` class.

(ii) It calls the global `display()` function and then calls the `display()`method of the `Printer` class.

(iii) The code results in a run time error.

(iv) The code results in a compiler error.

(e) State True or False. Unlike C, C++ allows comparing string objects directly using the == operator to check if their contents are equal. **(2 marks)**

(f) Which one of the following statements about `inline` functions in C++ is TRUE? **(2 marks)**

(i) `inline` functions cannot have any parameters.

(ii) `inline` functions can reduce function call overhead by directly inserting the function's code at the point where it is invoked.

(iii) `inline` functions must be declared as `static` if used in a class.

(iv) `inline` functions cannot have return types other than `int`.

(g) What will be the output of the following C++ program? **(2 marks)**

```
#include <iostream>

void process(int x) {
    std::cout << "process(int): " << x << std::endl;
}

void process(float x) {
    std::cout << "process(float): " << x << std::endl;
}

int main() {
    process('A');
    return 0;
}
```

(h) State True or False. In a C++ function, if a global variable and a local variable have the same name, the local variable always hides the global variable, and the global variable cannot be accessed within the same function. **(2 marks)**

(i) Which of the following statements accurately describes the behavior of the following C++ code segment. **(2 marks)**

```
#include <iostream>
class Sample {
    int value;
public:
    Sample(int v) : value(v) {}
    void setValue(int v) const {
        value = v;
    }
};
int main() {
    Sample s(10);
    s.setValue(20);
    return 0;
}
```

(i) The program compiles and runs successfully, setting the value as intended.

(ii) The program results in a runtime error because a `const` object cannot be modified.

(iii) The program results in a compiler error because `setValue()` is marked `const` but attempts to modify a data member.

(iv) The program compiles but `setValue()` has no effect because the object is `const`.

(j) Consider the following C++ namespace declarations. Write a C++ statement that invokes the `speed()` function of the `Physics` namespace with arguments 100 and 20, representing distance and time, and prints the result. **(3 marks)**

```
namespace Physics {
    double speed(double distance, double time) {
        return distance / time;
    } }
namespace MathUtils {
    double speed(double u, double a, double t) {
        return u + (a * t);
    } }
```

(k) What will be the output of the following C++ program? **(4 marks)**

```
#include <iostream>

class Example {
    static int count;
public:
    Example() {
        count++;
    }

    static void showCount() {
        std::cout << "Count: " << count << std::endl;
    }

    void resetCount() {
        count = 0;
    }
};

int Example::count = 0;

int main() {
    Example e1, e2;
    e1.showCount();
    e2.resetCount();
    Example::showCount();
    return 0;
}
```

3. **C++ Strings, Inheritance, Friend Functions** **(35 marks)**

(a) Write a C++ program that reads two words from the user and prints whether they are equal. Use appropriate C++ constructs (not C-style functions), and include only the `iostream` and `string` headers. **(5 marks)**

(b) What will be the output of the following C++ program? **(3 marks)**

```
#include <iostream>
#include <string>

int main() {
    std::string animals[3] = {"Cat", "Dog", "Elephant"};
    std::cout << animals[animals[0].length() - 1] << std::endl;
    return 0;
}
```

(c) What will be the output of the following C++ program? **(4 marks)**

```
#include <iostream>

class Base {
public:
    void show() {
        std::cout << "Base show" << std::endl;
    }
protected:
    void hidden() {
        std::cout << "Base hidden" << std::endl;
    }
};

class Derived : private Base {
public:
    void accessBase() {
        show();
        hidden();
    }
};

int main() {
    Derived d;
    d.accessBase();

    return 0;
}
```

(d) Answer the following questions. **(4 marks)**

i) What is a pure virtual function in C++? (2 marks)

ii) Given the function name `display()`, provide its declaration as a pure virtual function within a base class. (Only the single-line function declaration is expected; not a full class definition.) (2 marks)

(e) What will be the output of the following C++ program? **(4 marks)**

```
#include <iostream>
class Base {
public:
    Base() {
        std::cout << "Base default constructor" << std::endl;
    }
    Base(int x) {
        std::cout << "Base parameterized constructor: " << x << std::endl;
    }
};
class Derived : public Base {
public:
    Derived() {
        std::cout << "Derived default constructor" << std::endl;
    }
    Derived(int y) : Base(y) {
        std::cout << "Derived parameterized constructor: " << y << std::endl;
    }
};
int main() {
    Derived d1;
    Derived d2(10);
    return 0;
}
```

(f) What will be the output of the following C++ program? **(4 marks)**

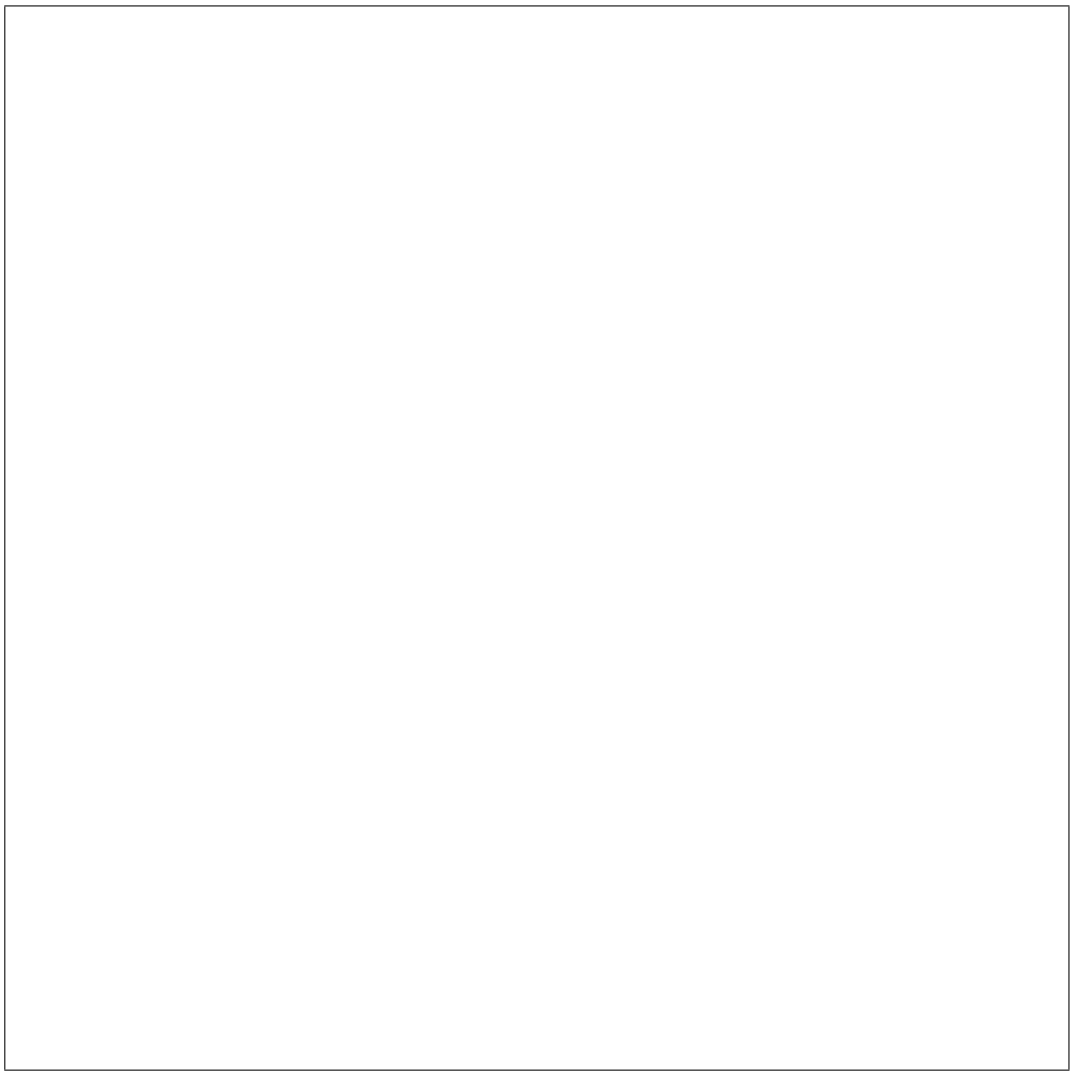
```
#include <iostream>

class Animal {
public:
    void speak() {
        std::cout << "Animal speaks" << std::endl;
    }
    virtual void sound() {
        std::cout << "Animal sound" << std::endl;
    }
};

class Dog : public Animal {
public:
    void speak() {
        std::cout << "Dog speaks" << std::endl;
    }
    void sound() {
        std::cout << "Dog barks" << std::endl;
    }
};

int main() {
    Animal* a;
    Dog d;
    a = &d;
    a->speak();
    a->sound();
    delete a;
    return 0;
}
```

(g) Consider the following C++ class declaration: **(7 marks)**

```
class Solid {
public:
    virtual float volume() const {
        return 0;
    }
};
```

Declare a new class named `Box` which extends `Solid` (assume the classes are in the same namespace). The `Box` class should have the following members:

- Two float member variables named `height` and `baseArea`, which should be accessible only to member functions and inherited by descendant classes.
- A public constructor that accepts two floats and initializes the `height` and `baseArea` members using an initializer list.
- A member function that overrides `volume()` and returns the product of `height` and `baseArea`.



(h) Consider the following C++ class. Provide the definition ofa friend function named `addDistances()` that takes two `DistanceMeters` objects as arguments and returns a new `DistanceMeters` object whose `meters` value is the sum of the two input objects. **(4 marks)**

4. **Structures, Containers, Templates** **(25 marks)**

(a) In C++, what is the default access specifier for the members of a structure (`struct`)? **(2 marks)**

(b) Which one of the following statements about structures in C++ is TRUE? **(2 marks)**

(i) A structure cannot have member functions.

(ii) A structure can only store data of the same type.

(iii) A structure can have constructors and member functions.

(iv) Structures in C++ are identical to structures in C.

(c) In a single statement, declare a `vector` named `v` containing 4 elements of type `float`, all default-initialized. **(2 marks)**

(d) State True or False. In C++, when using a function template, its type must always be specified explicitly by the programmer during each function call. **(2 marks)**

(e) State True or False. In C++, the `std::list` container provides direct access to its elements using the subscript operator []. **(2 marks)**

(f) In C++, the `std::set` container automatically maintains its elements in sorted order and does not allow duplicate values. **(2 marks)**

(g) Write a generic C++ function template named `swapValues()` that swaps the values of two variables of the same type. **(4 marks)**

(h) Which of the following statements about C++ containers is TRUE? **(3 marks)**

(i) `std::vector` and `std::list` both support random access using the [ ] operator.

(ii) `std::set` allows duplicate values.

(iii) `std::map` stores key-value pairs and automatically sorts the keys.

(iv) `std::vector` provides faster insertion and deletion at the beginning than std::list.

(i) State True or False. In C++, if you define a structure without any constructors, the compiler automatically provides a default constructor. **(2 marks)**

(j) What is the output of this C++ program? **(4 marks)**

```
#include <iostream>
#include <list>
#include <iterator>

using namespace std;

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

int main(void)
{
    int i;
    list <int> list1;
    for (i = 0; i < 8; ++i) {
        list1.push_back(i);
    }
    list1.reverse();
    list1.push_front(i);
    list1.pop_back();
    show(list1);
    return 0;
}
```

5. **File Handling and Dynamic Memory Allocation in C++** **(10 marks)**

(a) In a single statement, write C++ code to open a file named `data.bin` for reading in binary mode. Name the file stream object `fin`. **(2 marks)**

(b) Name the C++ class used for writing to files. **(2 marks)**

(c) Which operator is used to deallocate dynamically allocated memory in C++? **(2 marks)**

(d) Consider the following C++ statement. **(2 marks)**

```
int* ptr = malloc(sizeof(int));
```

This code will cause a compile-time error in C++. Rewrite the statement to make it valid in C++.

(e) Write a C++ statement that dynamically allocates an array of 10 integers using the `new` operator and stores the pointer in a variable named `arr`. **(2 marks)**

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

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