09:00–11:00 hrs	14 / June / 2024	Exams Hall	EXAMS OFFICE USE ONLY

University of the Witwatersrand, Johannesburg

Course or topic No(s)	COMS1018A, COMS1022A
Course or topic name(s) Paper Number & title	Introduction to Algorithms and Programming
Examination to be held during the month(s) of	June 2024
Year of study	
Degrees/Diplomas for which this course is prescribed	
Faculties presenting candidates	Science
Internal examiner(s)	Dr Steven James x-76157
External examiner(s)	Pula Rammoko/Dr Linda Marshall (UP)
Special materials	None
Time allowance	2 Hours
Instructions to candidates	50 Marks available. 50 marks = 100%. Answer all questions. This is a closed-book exam. This exam consists of 12 pages, including this cover page. Write your student number on each page.

Student Number: _____ Row: _____ Seat: _____

For marking purposes only

Question 1	
Question 2	
Question 3	
Question 4	
Total	

Instructions

- Please write your student number on each page, including the front cover.
- Answer all questions in pen. **Do not write in pencil.**
- Please write your student number at the top of each page.
- This exam consists of 12 pages. Ensure that you are not missing any pages.
- This is a **closed-book** exam: you may not consult any written material or notes.
- You are allocated 2 hours to complete this exam.
- There are 4 questions and 50 marks available.
- Ensure your cellphone is switched off.

Question 1 Short Questions [18 Marks]

1. What is the difference between the declaration and initialisation of a variable in C++? Provide an example of code to illustrate this difference. [2]

2. When a C++ program is run, the operating system must allocate memory in RAM for it. Which phase of the compilation cycle is responsible for this? [1]

3. In which part of memory are dynamically allocated variables stored? [1]

4. For each statement below, state whether it is true or false. [2]

   (a) Local, automatically allocated C++ variables are stored on the stack.

   (b) Indentation is semantically meaningful in C++ programs.

   (c) If the value returned by the main function of a C++ program is 0, it means an error has occurred.

   (d) If we wish to use smart pointers, then we must include the `<utility>` header file.

5. Rewrite the declaration of the following C++11 array to create the equivalent dynamic array that is allocated on the heap: `array<string, 10> words;` [1]

6. Explain what a "memory leak" is and how this can occur. [2]

7. The value of the letter `g` in the ASCII table is 103. What is the value of `d`? [1]

8. What header file should be included if we wish to use the built-in `sort` function? [1]

9. We are writing a program and wish to store the total amount of tax collected by the government in **cents**. There are roughly 10m taxpayers, each of whom pay about R40 000 on average in tax. What is the best built-in data type to use in this instance? Motivate your answer. [2]

10. Write a single line of code that deletes `x`, a dynamically-allocated array. [1]

11. Why might we prefer to pass a large object (such as a large `vector`) to a function by reference instead of by value? [1]

12. How many elements does a vector declared using `vector<int> v{17};` contain? [1]

13. What three elements make up a function's prototype in C++? [2]

Question 2 Code Flow [10 Marks]

1. Predict the output of the following code snippet. [1]

```
string s2 = "Hello";
string s1 = " World!";
cout << s2 << s1 << endl;
```

2. Predict the text that is displayed to the terminal when a user calls the function `f` in the code below. [2]

```
bool g(int x){
    if (x == 1){
        cout << "True" << endl;
        return false;
    }
    cout << "False" << endl;
    return true;
}
void f(){
    if (true && g(1)){
        cout << "True" << endl;
    }
    else{
        cout << "False" << endl;
    }
}
```

3. Predict the output of the following code that is executed on a machine where integers are represented using 32 bits and a memory address is 64 bits wide. [3]

```
int x = 4;
int arr [16] = {0};
int *y = &x;
cout << sizeof(*y) << " vs " << sizeof(&y) << " vs "
     << sizeof(arr)/sizeof(arr[0]) << endl;
```


4. Predict the output of the following code snippet. [2]

```
string a = "COMS";
int N = a.length();
for (int i = 0; i < N - 1; ++i){
    cout << i << ":";
    for (int j = i + 1; j < N; ++j){
        cout << " " << a[j];
    }
    cout << endl;
}
```

5. Predict the output of the following code: [2]

```
int arr [4] = {3, 4};
int *b = &arr[1];
cout << *b << endl;
b++;
cout << *b << endl;
int *c = arr + 3;
cout << c - &arr[1] << endl;
cout << (arr == &arr[1] - 1) << endl;
```

Question 3 Debugging [12 Marks]

1. The following code reads five integers into a vector, doubles each element and then outputs the new elements on a single line. However, there are **three errors** in the code that will prevent it from compiling. For each error, identify it by line number, explain why it is an error and rewrite the line to correct it. [6]

```
#include <iostream>
#include <array>

using namespace std;

int main(){
    int N = 5;
    array<int, N> arr;
    for (auto &element : arr){
        cin << element;
    }
    multiply(arr, 2);
    for (auto &element : arr){
        cout << element << ' ';
    }
    cout << endl;
    return 0;
}

void multiply(array<int, 5>& arr, int multiple){
    for (auto &element : arr){
        element *= multiple;
    }
}
```


2. The following function accepts a single integer N. It then reads in N integers from the user and outputs the minimum, maximum and mean values to the screen. However, the function occasionally outputs the wrong result. There are **two issues** inside the function that cause this incorrect behaviour. For each error, identify it by line number, explain why it is an error and rewrite the line to correct it. Note that the relevant header files, namespace, etc., are included but simply not shown here. [4]

```
void calculate(int N){
    vector<int> nums(N);
    sort(nums.begin(), nums.end());
    for (int i = 0; i < N; ++i){
          cin >> nums[i];
    }
    int min = nums[0];
    int max = nums[N-1];
    int sum = 0;
    for (int x : nums){
        sum += x;
    }
    double ave = sum / N;
    cout << "The min is " << min << endl
         << "The max is " << max << endl
         << "The mean is " << ave << endl;
}
```

3. Your friend, Richard, is learning to program and has written the following C++ function to calculate the sum of a `vector` of integers:

```
int summation(vector<int> numbers){
    int sum = 0;
    for (int i = 0; i <= numbers.size(); i++){
        sum += numbers[i];
    }
    return sum;
}
```

However, he is confused by something strange! Most of the time, the function returns the correct result, but very occasionally, it returns completely the wrong value. He asks for your help, and you tell him the issue is that his loop runs one too many times: he should instead have written `i < numbers.size()`. Richard understands he has made an error, but he does not understand why the error causes his program to sometimes work, and sometimes fail. Explain what is happening by making reference to the underlying memory that is being accessed by his incorrect loop. Your answer should explain what happens to make his function correct, and what happens when it is incorrect. [2]

Question 4 Programming [10 Marks]

1. Consider the function below:

```
vector<int> calculate(int N){
    int y = 2;
    vector<int> ans;
    while (y <= N){
        while (N % y == 0){
            ans.push_back(y);
            N = N / y;
        }
        y++;
     }
     return ans;
}
```

First, list the elements in the returned vector when the function is called with $N = 20$. Then, in plain words, explain the purpose of the function, and the meaning of the vector it returns. [3]

2. Consider the function below that accepts a `vector` of integers and a single non-negative integer `N`, and returns the sum of the first `N` elements of the vector:

```
int sumTo(vector<int> numbers, int N){
    int sum = 0;
    for (int i = 0; i < N; i++){
        sum += numbers[i];
    }
    return sum;
}
```

Rewrite this function in the space below so that it is instead **recursive**. Your function may not contain any loops. [3]

3. Complete the C++ program below by filling in the blank lines. The program reads an integer value from the user and determines whether or not this integer is an Armstrong number. An Armstrong number is an integer that is equal to the sum of each of its digits to the power of the number of digits in the number. An example of a three-digit Armstrong number is 407, because $407 = 4^3 + 0^3 + 7^3$. [4]

```cpp
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    int originalValue;
    cin >> originalValue;
    int currentValue = originalValue;
    int sum = 0;
    //calculate number of digits in the input:
    int numDigits = log10(abs(originalValue)) + 1;

    while (currentValue > 0){

        int digit = ____________________; //get the last digit

        sum = sum + ____________________;

        currentValue = ____________________; //drop last digit
    }

    if (_____________________________){
        cout << originalValue << " is Armstrong" << endl;
    }
    else {
        cout << originalValue << " is not Armstrong" << endl;
    }
    return 0;
}
```


This page intentionally left blank for extra space