

CSC3022: C++ Consolidation session 3

Question 1: Theory

Attempt to answer the following questions without running the code fragments. Then run the code fragments to check your answer.

Submission: You can use OpenOffice to answer these questions, and must submit a single PDF file, [student number].pdf, in the root folder of your tar file.

1.1. Give the output of the following code fragment:

```
int i=3;
cout << *&i << "\n";
```

Explain what the expression “*&i” does.

1.2. Explain what misinterpretation can be caused by the following declaration

```
int* p1, p2;
```

Correct this declaration to create two integer pointers p1 and p2.

1.3. Give the output of the following code. Explain how the statements at point 1 and point 2 affect the value of a in the main function.

```
int add(int a, int b){
    a = 5; // point 1
    return a+b;
}

int add(int* a, int* b){
    int x=*a;
    int y=*b;
    int sum = x+y;
    *a = 5; // point 2
    return sum;
}

int main(){
    cout << "a: " << a << " + b: " << b << " = "
        << add(a,b) << "\n";
    int a=1;
    int b=3;

    cout << "a: " << a << " + b: " << b << " = "
        << add(&a,&b) << "\n";
}
```

1.4. We have covered pointers and memory in lecture 5. This code also uses the new operator, which creates a dynamic variable on the heap. Consider the C++ program fragment below:

```
double average(int x, int y){
    // point 3
    return (x+y)/2.0;
}
```

```
int main(){
    int a=7;
    int b=2;

    int *intPtr1, *intPtr2;
    intPtr1 = &a;
    intPtr2 = &b;

    // point 1
    intPtr1 = new int;

    // point 2
    delete intPtr1;
    double d = average(a,b);

    // point 3
    intPtr1 = nullptr;
    intPtr2++;

    // point 4
}
```

- a) Complete the table below to show the memory allocation for all variables in the heap and stack at **points 1, 2, 3 and 4** in the program. A different set of tables should be completed for each point. You should use an appropriate function to determine the size of the different variable types. Assume that the memory addresses in the stack starts at 1000 and the heap at 5000 and memory is allocated sequentially as required. Show the allocated addresses, name, type and current value for each variable.

Stack

Address	Variable	Type	Value
1000			
1001			

Heap

Address	Variable	Type	Value
5000			
5001			

- b) Change the average function header to use pointer parameters. Show the new memory usage at point 3.
- c) Consider the modified program below:

```
double average(int x, int y){
    double *dPtr = new double;
    *dPtr = (x+y)/2.0;
    return *dPtr;
}

int main(){

    int j=20;
    for(int i=1; i<=5; i++){
        cout << average(i,j) << "\n";
    }

    // point 1
}
```

Note that C++ does not have a garbage collector like Java. This method introduces a memory leak. Explain why this happens by showing the memory usage on the heap at point 1.

Question 2: Practical

2.1. Write a C++ program which performs a simple test to determine if a string or a sentence is a palindrome, i.e. the given sentence is the same if it is read backwards. Your program must define and call an appropriate function, called `reverseStr`, which reverses a given string. You should consult the reference manual on cplusplus.com to find appropriate functions and operators for the `std::string` class.

Some example output is given below.

```
Enter a sentence below to check (-1 to end):
civic
civic reversed is: [civic]
civic is a palindrome

Enter a sentence below to check (-1 to end):
hello
hello reversed is: [olleh]
hello is not a palindrome

Enter a sentence below to check (-1 to end):
radar
radar reversed is: [radar]
radar is a palindrome

Enter a sentence below to check (-1 to end):
12345 54321
12345 54321 reversed is: [12345 54321]
12345 54321 is a palindrome
```

Answer this question in a single `main2_1.cpp` file. Submit a single tar file which contains both your PDF and the `.cpp` file.

2.2. This is an optional exercise (`main2_2.cpp`)

The program above is case sensitive and also takes into account punctuation marks, e.g. the sentence “Madam, I'm Adam” will not be detected as a palindrome. Write a function, called `stripNonAlphaChars`, which strips non alphabetic characters from a string and converts the string to lowercase. **HINT:** You must investigate and use appropriate function(s) from the standard C++ localization library.

```
Enter a sentence below to check (-1 to end):
```

```
Madam, I'm Adam
```

```
Madam, I'm Adam stripped is: [madamimadam]
```

```
Madam, I'm Adam reversed is: [madamimadam]
```

```
Madam, I'm Adam is a palindrome
```

```
Enter a sentence below to check (-1 to end):
```

```
Was it a car or a cat I saw?
```

```
Was it a car or a cat I saw? stripped is: [wasitacaroracatisaw]
```

```
Was it a car or a cat I saw? reversed is: [wasitacaroracatisaw]
```

```
Was it a car or a cat I saw? is a palindrome
```

```
Enter a sentence below to check (-1 to end):
```

```
Hello there, Theresa
```

```
Hello there, Theresa stripped is: [aserehterehtolleh]
```

```
Hello there, Theresa reversed is: [hellotheretheresa]
```

```
Hello there, Theresa is not a palindrome
```