

Week 3 Quiz

| | | | |
|------|------------|------|---------|
| Date | 24/02/2023 | Time | 15 mins |
|------|------------|------|---------|

1. (12 pts each, 60 pts total) Answer the following questions.

(a) Is the following statement true or false? Give reasons for your answer.

We can deallocate dynamic memories and delete pointer variables using the delete operator.

→ *False. delete operator just be used for deallocating dynamic memories, not for deleting pointer variables.*

(b) What is a segmentation fault?

→ *Segmentation fault is a specific error caused by accessing memory that is not yours.*

For examples, you have dereferenced the null pointer; you have stepped outside the array bounds; or you are accessing memory that has already been deallocated.

(c) Write a code to allocate memory for an array of characters of size `n` and assign the address of this memory to pointer `charPtr`.

→ `char *charPtr = new char[n];`

(d) Given `pi` is a pointer to an array of integers of size 25 (`ai`). State the meaning of the following statement: `*(2 + pi) = 24;`

→ *The above statement stores 24 into the 3rd element of the array ai. It is equivalent to `*(pi + 2) = 24;` and `ai[2] = 24;`.*

(e) How to display the address of a pointer of type `char` (with `cout`)?

→ *We can typecast the pointer to another pointer type, such as `(int *)`.*

2. (10 pts each, 20 pts total) Given a `Student` struct having the following declaration. Fulfill the following requirements.

```
struct Student
{
    char name[51];
    char ID[10];
    float grade;
};
```

(a) Given `Student *ptrStu = new Student;`. How would we access the name of the student `ptrStu`?

→ `ptrStu->name;` or `(*ptrStu).name;`

(b) Given `Student *ptrArr = new Student[5];`. How would we access the ID of the second student of the `ptrArr`?

→ `ptrArr[1].ID;`

3. (20 pts) In any case, what is the output from the program below? Explain your answer.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int arr[5] = {2, 4, 6, 8, 10};
7      int *ptr = arr;
8
9      while (ptr < &arr[4])
10     {
11         ptr++;
12     }
13
14     while (ptr > arr)
15     {
16         cout << *ptr << " ";
17         ptr--;
18     }
19
20     return 0;
21 }
```

→ The output is 10 8 6 4 .

Explanation:

- Initially, `ptr` is a pointer to the first element of the `arr` (line 7).
- Next, after completing the first while loop (line 9), `ptr` points to the last element of the array `arr`.
- Finally, for each iteration in the second while loop (line 14), the value of the memory address that the `ptr` points to is printed out, and then `ptr` decreases. The while loop is repeated until `ptr` points to the first element in the `arr`. Because `ptr` equals `arr` → the while loop ends → the value of the first element is not printed.

END
