Object Oriented Programming & Design - Monsoon 2023

Midterm Practice Questions

Student's Details

Name of the Student:

Roll Number:

Stream:

Question Structure and Instructions

This paper consists of two sections:

1. The 1st section consists of 10 short conceptual questions each carrying 3 points.
2. The 2nd section consists of 2 longer questions each carrying 5 points.

1 Short Conceptual Questions (3x10=30 points)

There are 10 questions in this section each worth 3 points. You may write the answers to these questions succinctly.

A1 Consider the following program. Specify and then explain its output, while specifying whether it correctly swaps.

```
int swap(int arr[]) {
    int temp = arr[0];
    arr[0] = arr[1];
    arr[1] = temp;
}

int main(void) {
    int array = [4, 6];
    swap(array);
    cout << array[0] << " " << array[1];
    return 0;
}
```

A2 Suppose we need to perform a single arithmetic operation on all elements of a data structure. Using which data structure between array and linked list will this be faster and why?

Roll Number:

A3 Suppose we build a C++ program without using any flags once, and measure the time taken to build. We again re-build the same program while switching on the flag 'O2', and again measure the time taken. Which of them would take higher amount of time and why?

A4 Consider a case of a program where every data member and member function is public. Which of the principles of object-oriented programming is NOT satisfied by this design?

A5 Consider the following code fragment:

```
class c {
    public:
    int *arr;
    c() { arr = new int[10];
        arr[0] = 1;
    }
};

int main(void) {
    c *obj = new c;
    obj->arr[1] = 2;
    delete obj;
    obj = new c;
    obj->arr[2] = 3;
    return 0;
}
```

Is there any problem with the above code? If so what is it?

A6 Consider the following program fragment:

```
class c {
    public:
    int arr[10];
    c() { arr[0] = 1; }
};
class d: public c {
    public:
    int arr2[10];
    d() { arr2[5] = 1; }
};

int main(void) {
    c *obj = new c;
    obj->arr[1] = 2;
    delete obj;
    d *obj = new d;
    obj->arr[2] = 3;
    cout << obj->arr2[5] << " " << obj->arr[2];
    return 0;
}
```

What is the output of the above program and why?

A7 Suppose we want to keep track of the students of the OOPD course. The OOPD course requires grading of the students who are taking 2 credits separately from the ones who are taking 4 credits. Assume that a function *assignGrade* within

the *Student* class is specified to compute the grade, which technique needs to be used within the *Student2Credit* and *Student4Credit* classes to ensure that grades properly assigned? Show using a basic code structure.

A8 Suppose we decide to implement another functionality of removing a student from the class in case somebody drops out. Is it possible to utilize the minus (-) operation to do it? If so, what is the technique, and illustrate it with a fragment of code.

A9 Consider a library function that implements sorting of an array of both integers and floats. A key feature of the sorting of floats is that values that do not represent a number (such as infinity, -infinity, not a number) are kept at the end of the array. Which feature in C and C++ (separately) can be used to implement this scenario, if you want to avoid an if-else condition in the main function?

A10 Consider a class which has a copy constructor, but it uses call-by-value. Is it fine to use it, or would it lead to an error? Justify.

2 Long Questions (5 x 2 = 10 points)

There are 4 questions in this section each worth 5 points.

B1 Design the class structure, with specifications of which functions should be virtual, of a program that keeps track of the rooms in the R&D Building of our campus. The rooms have three distinct types – offices, classrooms and labs. Offices and labs are assigned to some faculty, whereas classrooms are not. Classrooms have different capacities and other facilities, like projectors, boards and so on.

B2 Arrange the following functions in increasing order of their speeds.

```
void fun_0(int arr[], int n) {
    for (int i = 0; i < n; i++) {
        arr[i] = arr[i] * 2;
    }
void fun_1(float arr[], int n) {
    for (int i = 0; i < n; i++) {
        arr[i] = arr[i] * 2.0;
    }
}
void fun_2(float arr[], int n) {
    for (int i = 0; i < n; i++) {
        arr[i] = arr[i] + 2.0;
    }
}
```