Object Oriented Programming & Design - Monsoon 2024

Midterm Examination Questions

Name of the Student:

Roll Number:

Stream:

Question Structure and Instructions

Please write the **set number** on the first page of the answer sheet. 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 4 longer questions each carrying 5 points.

Set-C

1 Short Conceptual Questions (3x10=30 points)

A1 Consider a program with a time complexity of $O(n)$, where n is the length of the input array. Running this program with an array of 100 elements gives an execution time of around 1 microsecond. An array of 1000 elements takes 5 microseconds. How much would an array of 10000 elements take? If it is not possible to compute, can you compute what is the minimum time it would at least take, i.e. a lower bound on the time?

(a) Cannot exactly compute – 2 points

(b) 45 microseconds; need to construct a straight line, find their slopes and intercepts – 1 point

A2 Suppose there is a github repository, on which users A and B are both working. Both of them clone the repository, then make changes between Lines 100-150 of the program. User A first commits and then pushes the changes. User B then tries to commit and then push his/her own changes. What would be the output of commit and push of each user? If any problem occurs, what is the fix?

(a) Output of commit of first user – Success – 0.5 points

(b) Output of push of first user – Success – 0.5 points

(c) Output of commit of second user – Success – 0.5 points

(d) Output of push of second user – Conflict (1 point); need to identify the conflict cases and manually merge (0.5 point)

A3 Suppose you have a program that requires you to use bubble sort, binary search and a matrix multiplication, each with time complexity $O(n^2)$, $O(logn)$ and $O(n^3)$. During testing, a developer finds it to be very slow, and decides to replace the traditional technique of matrix multiplication by a more optimized algorithm, called Strassen's algorithm. Is this the right approach or not? If this is the right approach, explain why. If this is the wrong approach, then what is the right approach and why? Would any tools be necessary?

No, this is not the right approach, since a high time complexity need not always imply that it has high execution time. The right approach is to utilize a profiler like gprof to find out which part of the code consumes most time.

(a) Not the right approach + time complexity need not imply high execution time – 1 point

(b) Right approach is to use a profiler/gprof to identify the portion of code consuming most time – 2 points

A4 Consider the following code:

```
File p1.cpp
int a = 0;
void fun();
int main(void) {
    a = 15;
    fun();
    std::cout << a;
}

File p2.cpp
int a = 5;
void fun() {
    a = a * a * a;
}
```

Now consider the following shell commands to build the above code:

```
>gcc -c p1.cpp
>gcc -c p2.cpp
>gcc p1.o p2.o
```

What will be the output of the above commands and which files would be generated after each command?

(a) No output; p1.o will be generated – 1 point

(b) No output; p2.o will be generated – 1 point

(c) Linking error / error – 1 point

A5 Consider the case of the following program with dynamic memory allocation:

```
int main(void) {
    int *ptr = NULL;
    ptr = new int[100];
    ptr[0] = 25;
    ptr[10] = 27;
    ptr[20] = 100;
    std::cout << ptr[0] << *(ptr + 10) << *ptr + 20;
    ptr = new int;
    *ptr = 3;
    delete ptr;
    return 0;
}
```

Will the above program compile and then run properly? If so, what will be its output? Are there any problems in the code and if so, what are they?

(a) Yes, the program will compile and run – 1 mark

(b) Output – 25 27 45 (space or no space are both okay) – 1 mark

(c) Yes, the above program leads to a memory leak – 1 mark

A6 What will be the output of the following code?

```
void swap1(int a[]) {
    int c;
    c = a[1];
    a[1] = a[0];
    a[0] = c;
}
void swap2(int &a, int &b) {
    int c;
    c = b;
    b = a;
    a = c;
}
int main(void) {
    int a[] = {6, 5};
    swap1(a);
    swap2(a[0], a[1]);
    std::cout << a[0] << " " << a[1] << std::endl;
    return 0;
}
```

(a) 6 5 – 2 marks

(b) Both the swaps work as swap1 sends the whole array, whereas swap2 uses call by reference – 1 mark (if justification is not written then also don't cut marks, if answer is correct give full marks)

A7 Write the output of the following code with justification:

```
class Base {
    public:
    Base() { cout << "Base Function Constructor"; }
    void base() { cout << "Base's base()"; }
};
class Derived: public Base {
    public:
    void base(int a) { cout << "Derived's base(int)"; }
};
int main(void) {
    Derived *d = NULL;
    d = new Derived;
    d->base();
    delete d;
    return 0;
}
```

(a) Output: Base Function ConstructorBase's base() – 1 + 1 marks

(b) This is a case of function overloading, since there are two versions of the base() function – 1 mark

A8 Write the output of the following code with justification:

```
int add(int a, float b) { return a + b + 10.5; }
int add(int a, int b) { return a + b; }
int main(void) {
    std::cout << add(40, 50) << " " << add(4, 5.0);
    return 0;
}
```

(a) 90 19 – 1.5 marks

(b) The first call goes to the one that takes integer arguments, second call goes to the one that takes float arguments (only if first answer is correct) – 1.5 marks

A9 Write the output of the following code with justification:

```
int main(void) {
    try {
        for (int i = 0; i < 10000; i++) {
            if ((i % 500) == 0)
                throw throw std::logic_exception{"unexpected"};
            cout << i;
        }
    } catch (exception e) {
        cout << "Logical error";
    }
}
```

(a) Logical error – 2 points

(b) The first iteration of the loop itself raises an exception and execution no longer happens (only if first answer is correct) – 1 point

A10 Consider a class which has two distinct types of constructors. One constructor uses static memory allocation, whereas the other constructor uses dynamic memory allocation. The decision about which one to use is made using function overloading. How would you ensure that the destructor of that class can handle both cases? You should explain with an example.

(a) There can be only one destructor, which needs to check if static or dynamic memory allocation was used – 1.5 marks

(b) Example of destructor – 1.5 marks

2 Long Questions (5 x 4 = 20 points)

B1 Consider the way a processor chip can be modelled. A single processor chip contains a number of cores, as well as L1 and L2 cache memory. Each core has its own set of registers. Registers are of two types – special-purpose, for the internal use of the processor's logic, and general-purpose, for storing data of the programs. The processor chip also has additional properties such as the maximum frequency, number of bits in each register and the instruction set architecture. Design a class structure to model a processor chip.

```
class Processor {
    Core *cores;
    int numOfCores;

};
class Register {
    string name;
};
class Core {
    Register *register;
};
class Register {
    string name;
```

```
};
class GeneralPurposeRegister: public Register {
};
class SpecialPurposeRegister: public Register {
};
```

(a) 0.5 points each for the 5 class – 2.5 (this should not be done by inheritance)

(b) 0.5 points for having cores pointer in Processor class – 0.5

(c) 0.5 points for register pointer in Core class – 0.5 (inheritance should not be used here)

(d) 1.5 points for using inheritance with Registers

B2 Consider the following programs:

P1.cpp

```
 int x[2000], y[2000], z[2000];
for (int i = 0; i < 2000; i++) {
    z[i] = x[i] + y[i];
}
```

P2.cpp

```
float x[1000], y[1000], z[1000];
for (int i = 0; i < 1000; i++) {
    z[i] = x[i] + y[i];
}
```

(i) Which among P1.cpp and P2.cpp is faster and why? (ii) Can using the "-O2" flag speed up their execution, and why (not)? (iii) Can a GPU speed up their execution, and why (not)?

(a) P1 is faster, because even though P2 has smaller length of array by 2 times, addition of floats is slower than addition of ints by greater than 2 times – 1 + 1

(b) No, use of -O2 flag cannot speed up execution, because compiler optimization cannot speed up execution in this case. – 1 + 0.5

(c) Yes, a GPU can speed up execution, since it can exploit data-level parallelism – 1 + 0.5

B3 A single lab in an academic department consists of one or more faculty, one or more PhD students, one or more MTech students and one or more BTech students. All students have roll numbers. Occasionally, there are a few short-term employees also hired. A physical space is also assigned to each individual lab. Design the class structure to represent each lab of a department.

```
class Lab {
    Faculty *faculty;
    Student *student;
    Employee *employee;
    int roomnum;
};
class Student {
    string / int rollnum;
};
class BTechStudent: public Student { };
class MTechStudent: public Student { };
class PhDStudent: public Student { };
class Employee { string / int employeeId; };
```

(a) 0.5 marks each for 6 classes

(b) 1 mark for correctly mentioning faculty, student and employee data members in Lab class

(c) 1 mark for using inheritance with Student

B4 Delhi metro consists of a total of 10 lines and 256 stations, with each line containing a number of stations. Occasionally, a station also belongs to a number of lines. Each station also contains a number of platforms, which indicate the source and destination of the trains. Design a class structure to model the above, and mention in text or comments what type of relationship you are trying to model for each case.

```
class Station {
    string name;
    Platform *platforms; // C1: Composition; since platform can only exist if
};
class Platform {
    Line line; // C2: Aggregation: Each platform contains one line
    int pnum;
};
class Line {
    int linenum;
    Station *stations; //C3: Association: Lines are associated with stations
};
```

(a) Station class with platforms pointer – 1 mark

(b) Platform class with line variable – 1 mark

(c) Line class with stations (must use a pointer) – 1 mark

(d) Comment C1 – 1 mark

(e) Comment C2 – 0.5 marks

(f) Comment C3 – 0.5 marks