

OOP MID TERM

MCE 07905536, Shyed Shahriar ①

Ans. to q no-1

Difference between class & structure:

Point	Class	Structures
Type	Reference type	Value type
Inheritance	Supports	Do not supports
Memory Allocation	Heap (garbage collected)	Stack, part of containing object.
Nullability	Can be set to Null	Can not be null
Copy	Reference the same object	Operates on copy of data
Usages	Complex objects, inheritance, higharchies	Lightweight, efficiency performance, immutability.

Struct a { int i, j, k };

Class a { int i, j, k };

The difference between those

two are, the structure has all its members public by default (P.T.O)

OOP Midterm

MCE 07905536, Shyed Shahzad ②

While the class has all its members private by default, the struct can be or accessed & modified by any code but the class can not be accessed.

The class declaration is useless because it does not have any method or constructor to access or use the members. The class is basically a struct with hidden members.

```
class a{ public:  
    int i,j,k};
```

this will make the class & its members accessible by code just like the struct a.

MCE 07905536, Shyed, OOP mid ③

Ans. to q no- 2

Constructor in C++:

A constructor is a special type of member function in C++ that is called automatically when an object of its class is created. It is used to initialize the data members of the object.

There are three main type of constructors,

1) Default Constructors: A constructor with no parameters (takes no argument). If no constructor defined C++ automatically provides constructor and initialize with default values.

class Point { public:

 Point () { x=0; y=0; }

private:

 int x;
 int y;

};

MCE07905536, Shyed

(4)

2) Parameterized constructors: A constructor with parameters, preferred method to initialize members.

```
class Wall { private:
    double length;
    double height;
public:
    Wall (double len, double hgt)
    {
        length = len;
        height = hgt;
    }
};
```

3) Copy constructor: Used to copy data of one object to another.

```
class Wall { private: double length;
                double height;
public: Wall (double len, double hgt)
{
    length = len; height = hgt;
}
Wall (Wall &obj) { length = obj.length;
                    height = obj.height;
}
};
```

MCE 07905536, Shyed

OOP

5

Apart from ~~Ans. to q.~~, there are several other type of constructors, like - Dynamic constructor, conversion constructor, explicit constructor, move constructor etc.

Ans. to the q. no-3

A ~~o~~ class is a blueprint or template of creating objects, that define a set of attribute & methods. It is a user defined data type, encapsulating data & functions in one unit.

```
class ClassName{ // Data member
    public: int x; int y;
    // Method
    void set( int x, int y ) { z = x + y; }
    int get() { return z; }
}
```

class ClassName defines the class

(P.T.O)

MCE07905536, Shyed

(6)

x,y are data members (attributes)
set() & get() are member
functions (methods)

Object is an instance of a class.
Syntax for creating an object in C++

ClassName Objectname;
ClassName is class & Objectname is
object names respectively.

Objectname has access to all
the data & methods of the class
for example

Objectname.set(10,20)

~~set~~ sets x & y to 10 & 20

and int result=Objectname.get();

gets the result= (z=x+y)

MCE07905536, Shyed

(7)

Ans. to the q. no (4)

Example code to create a class AreaCalculate & use overloaded function Area to calculate the area of circle, rectangle, square & triangle
include <iostream>
using namespace std;

```
class AreaCalculate { public:  
    void Area (int r) { cout << "circle" << 3.14 * r * r  
                           << endl; }  
    void Area (int l, int b) { cout << "rectangle" <<  
                           l * b << endl; }  
    void Area (float base, float height) {  
        cout << "Triangle" << 0.5 * base * height <<  
                           endl; }  
    void Area (float a) { cout << "Square" <<  
                           a * a << endl; }  
    int main () { AreaCalculate AC ;  
        AC.Area (5); AC.Area (5, 10); AC.Area (7.0, 11.0);  
        AC.Area (9.0); return 0; }
```

Class

~~#~~ Areacalculate was used to create AC object. Area function method was overloaded and the result will be returned based on type and number of parameters & arguments.

Ans. to q.no-(5)

Example of creating a class named Student with data members to store roll, name, three subjects, max marks, min mark, obtained marks with input & display options. The code will use separate objects (of same class) for separate

students.

```
include <algorithm>
```

```
include <iostream>
```

```
include <std>
```

```
using namespace std;
```

```
class Student { public: int roll;  
string name; string sub1; }
```

MCE 07905536, Shyek

oop
⑨

```
string sub2; string sub3;
int max_marks; int min_marks;
int obtained_marks;

void input() {
    cout << "Enter roll:" << endl;
    cin >> roll;
    cout << "Enter name:" << endl;
    cin >> name;
    cout << "Enter Subject 1:" << endl;
    cin >> sub1;
    cout << "Enter Subject 2:" << endl;
    cin >> sub2;
    cout >> "Enter Subject 3:" << endl;
    cin >> sub3; }

void display() {
    cout << "Roll:" << roll
        << endl;
    cout << "Name:" << name << endl;
```

MCE07905536, Shyed

(10)

```
cout << "Subject 1 mark: " << sub1  
    << endl;  
cout << "Subject 2 mark: " << sub2  
    << endl;  
cout << "Subject 3 mark: " << sub3  
    << endl;  
  
cout << "Maximum Mark: " <<  
    max(max(sub1, max(sub2, sub3)),  
        sub3)) <<  
    endl;  
cout << Minimum Marks <<  
    min(min(sub1, sub2), sub3) <<  
    endl;  
cout << "Obtained Marks" <<  
    (sub1 + sub2 + sub3) << endl;  
}  
};
```

int main(){ Student s1 ;
 s1.input();
 s2.display(); return 0; }