



Course Code: CS-217	Course Name: Object-oriented Programming
Instructors: Dr. Abdul Aziz & Syed Zain Ul Hassan	
Student ID:	Section:
Date: September 23, 2019	Time: 60 minutes

Instructions:

Attempt all tasks. All questions carry equal marks

Return the paper after exam

There are 3 questions on 2 pages

Max Points: 35

Question 1: (2+2+2+2+2)

Give brief answers of the following:

a) How is **this** operator useful?

A: this operator refers to current object. It can be used to pass current object as parameter as well as for referring current class instance variables.

b) List any three situations where inline functions do not work.

A: 1) When function contains loops.
2) When function is recursive.
3) When a function contains switch/goto statements.

c) Why are member initialization lists useful?

A: It can be used to initialize constant instance variables with values taken at runtime.

d) In the presence of a parameterized constructor, it is not necessary to have a setter function. Do you agree with this statement? Justify your answer.

A: A parameterized constructor can be used to initialize instance variables but to later modify them program it is necessary to provide mutator (setter) functions.

e) How can we stop objects from being copied in our program?

A: By defining our own copy constructor in the class and making it private.

Question 2: (10)

Consider the following class definition:

```
class course {  
    string c_name, c_id;
```

```

        int c_hours;
        float course_gpa;
};

class student {
    string s_name, s_id;
    float s_gpa;
    private: void calculateSGPA() { }
};

```

- Make **s_id** constant and then implement constructors for **course** and **student** classes.
- Implement setter functions for class members that can be changed.
- Call **calculateSGPA()** from **main()** for a single student instance. Make any changes or add new class members if necessary to calculate the SGPA. The SGPA can be calculated as:

$$\text{SGPA} = (\text{Total GPA of all five courses in semester} / \text{Total Credit Hours attempted})$$

```

class course
{
    string c_name, c_id;
    int c_hours;
    float course_gpa;

    public:
    course()
    {
    }

    course(string c_name, string c_id, int c_hours, float course_gpa)
    {
        this->c_name = c_name;
        this->c_id = c_id;
        this->c_hours = c_hours;
        this->course_gpa = course_gpa;
    }

    void setCName(string c_name)
    {
        this->c_name = c_name;
    }

    void setCid(string c_name)
    {
        this->c_id = c_id;
    }

    void setChours(string c_name)
    {
        this->c_hours = c_hours;
    }
}

```

```

    }

    int getChours()
    {
        return c_hours;
    }

    float getCgpa()
    {
        return course_gpa;
    }
};

class student
{
    string s_name;
    const string s_id;
    float s_gpa;

private:
    void calculateSGPA(course courses[])
    {
        float totalGPA = 0;
        int totalHours = 0;
        for(int i = 0; i < 5; i++)
        {
            totalGPA += courses[i].getCgpa();
            totalHours += courses[i].getChours();
        }

        float s_gpa = totalGPA / totalHours;
        cout << "SGPA: " << s_gpa;
    }

public:
    student(string s_name, string id, float s_gpa): s_id(id)
    {
        this->s_name = s_name;
        this->s_gpa = s_gpa;
    }

    void setName(string s_name)
    {
        this->s_name = s_name;
    }

    void callCalculateSGPA(course courses[])
    {
        calculateSGPA(courses);
    }
}

```

```
};

int main()
{
    student s1("Ali", "K163058", 4.5);

    course c1("COAL", "ee213", 3, 2.6);
    course c2("OOP", "cs217", 3, 3.3);
    course c3("DB", "cs350", 3, 4.0);
    course c4("DLD", "cs105", 3, 3.3);
    course c5("ENGLISH", "cs101", 3, 4.0);

    course c[] = {c1, c2, c3, c4, c5};

    s1.callCalculateSGPA(c);
}
```

Question 3: **(15)**

Consider the following scenario:

"DreamCo is one of the leading IT companies of the country. The company has its head office located in Karachi, while it has other offices at Lahore, Islamabad & Hyderabad. The company provides three basic services namely Network Security Testing, Desktop Application Development and Website Development. There may also be other rival companies in the market that offer the same services as well.

For each of the three services, DreamCo has a separate department that has several employees and is managed by a Department Head. Although all of DreamCo employees get handsome salaries (Rs. 20,000 being minimum), the employees working in Network Security Testing are considered more valuable and are provided with an allowance apart from getting their monthly salary.

Keep in mind that each of the offices has these three departments.

In order to make business decisions, the owners sometimes like to see details about the three departments across all the offices i.e. the total number of employees working in each department and the entire amount being spent on salaries and allowances."

- a) Identify all the classes and objects that will be needed.
- b) Identify the key attributes and functionalities of each class.
- c) Draw an object relation/interaction model.
- d) Assume DreamCo inaugurates a new office in Faisalabad and wants it to initially follow the Lahore office as a blueprint. Add a copy constructor to create this duplicate office object.
- e) Explain how Object Oriented Programming approach can make your program more robust, secure and efficient with reference to the given scenario.

```
#include <iostream>
using namespace std;
```

```
class Employee
{
    string emp_name;
    int emp_salary;
    int emp_allowance;

    public:
    Employee () { }

    string getEmpName()
    {
        return emp_name;
    }

    void setEmpName(string emp_name)
    {
        this->emp_name = emp_name;
    }
};
```

```
class Department
{
    int dept_id; // ID: 1 for Network Security, 2 for Desktop apps & 3 for Web apps
    int no_of_employees;
    Employee dept_head;
    bool allowance_applicable = false;

    public:
    static int empCount;

    Department() { }

    Department(int dept_id)
    {
        if(dept_id == 1)
        {
            this->dept_id = dept_id;
            no_of_employees = 15;
            dept_head.setEmpName("Ali");
            allowance_applicable = true;
        }
    }
};
```

```

    }
    else if(dept_id == 2)
    {
        this->dept_id = dept_id;
        no_of_employees = 20;
        dept_head.setEmpName("Abid");
    }
    else
    {
        this->dept_id = dept_id;
        no_of_employees = 25;
        dept_head.setEmpName("Asad");
    }

    empCount += no_of_employees;
}

void showDeptSummary()
{
    cout << "Department ID: " << dept_id << endl;
    cout << no_of_employees << " work in this department." << endl;
    // Do the same as no. of employees for salary
    cout << "Mr/Ms. " << dept_head.getEmpName() << " is the head of this department." <<
endl;
}

int getNoOfEmp()
{
    return no_of_employees;
}
};

int Department::empCount = 0;
// another static variable for total salary

class Office
{
    string loc_city;
    Department * d;

public:
    Office() { }
    Office(string loc_city)
    {
        this->loc_city = loc_city;
        d = new Department(1); // a department for Network security
        d = new Department(2); // a department for desktop apps
        d = new Department(3); // a department for web apps
    }

```

```

    }

    void showOfficeSummary()
    {
        cout << "The " << loc_city << " has three departments." << endl;
        cout << "The total no. of employees working in this office is " << d->empCount << endl;
        // Do the same as total employees for total salary
    }

};

int main()
{
    Office o1("Karachi"); // ID: 1 for Network Security, 2 for Desktop apps & 3 for Web apps
    o1.showOfficeSummary();
}

```