

CS-217: Object Oriented Programming

Serial No:

Sessional Exam 2

Total Time: 1 Hour

Total Marks: 40

Wednesday, 25th November, 2020

Course Instructors

Jawad Hassan, Abdul Waheed khan, Subhan Ullah

Signature of Invigilator

Student Name

Roll No.

Section

Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt on question paper. Attempt all of them. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work. Use the back of the last page for rough work.
3. If you need more space write on the back side of the paper and clearly mark question and part number etc.
4. After asked to commence the exam, please verify that you have twelve (12) different printed pages including this title page. There are a total of 3 questions.
5. Calculator sharing is strictly prohibited.
6. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.

	Q-1	Q-2	Q-3	Total
Marks Obtained				
Total Marks	10	15	15	40

Question 1 [1+1+1+5+2 Marks]

(1-a): Given a structure definition, write a single line code in the space provided to access and print emName inside main function:

```
struct Employee {  
    int ID;  
    string emName;  
    string emSex;  
    string emAddress;  
    int emSalary;  
    Employee(int id, string name)  
    {  
        ID = id;  
        emName = name;  
    };  
  
void main()  
{  
    Employee* Emp = new Employee;  
  
    cout<<Emp->emName;  
}
```

(1-b): Given the following code, write a single line code in the space provided using the “record” variable of type **BioData** to change the lastName from “Khan” to “Kamal”.

```
struct Name_Info
{
    string firstName;
    string lastName;
};
struct BioData
{
    int ID;
    Name_Info* name = new Name_Info;
    string address;
}
void main()
{
    BioData record = { 123, {"Ahmad","Khan"},"House No. 10, XYZ Road" };
    record.name->lastName = "Kamal";
}
```

(1-c): Identify and correct any errors in the following code snippet. Also write the output produced.

```
struct Book
{
    int ID;
    string author;
    double price;
};
void main()
{
    Book Learn_Cpp[3]={{123,"Dietel"},"Toni",200},{789,,500}};

    Book Learn_Cpp[3] = {{123, "Dietel"},"456","Toni",200},{789," " ,500}};

    Book* ptr = Learn_cpp;
    cout << Learn_Cpp[0].price; // output: 0
    ptr = ptr + 1;             // output: Toni
    cout << ptr->author;
}
```

(1-d): Suppose you want to keep record of all members of a family. Naturally, all the family members should have a common family name. In this regard, given the following code, identify and correct any errors. Also, write the output produced.

Note: you are not suppose to make any changes to the code inside main function.

```
class Family_Member
{
    string givenName;

    char gender;

    static string family_name = "Abbasi";

    static int count = 0;

public:

    Family_Member(string name, char gen, string fName)
    {

        givenName = name;

        gender = gen;

        family_name = fName;

        count++;

    }

    static void printCount ( )
    {

        cout<<"Family Name is: "<<this->family_name<<endl;

        cout<< "total count of family members = "<<count;

    }

};

string Family_Member::family_name = "Abbasi";
int Family_Member::count = 0;
void main()
{
    Family_Member m1("Ahmad", 'M', "Khan"), m2("Sara", 'F', "Khan");
    m1.printCount();
}
```

Output produced: **Family Name is: Khan**
 Total count of family members = 2

(1-e): Given the following code for Employee class, **write an operator overloading function for '+' operator as a non-member function** that should provide the same functionality as the **addTwo ()** function.

```
class Employee
{
private:
    int idNum;
    double salary;

public:
    Employee(int id, double salary);

    double Employee::addTwo(Employee& emp)
    {
        double total;
        total = this->salary + emp.getSalary();
        return total;
    }
};
```

Solution

```
double operator + (Employee& emp1, Employee& emp2)
{
    double res = emp1.getsalary() + emp2.getsalary();
    return res;
}
```

Question 2 [6+4+5]

(2-a): Re-write the following program by separating the interface from implementation. You must create three separate files (Header file, .cpp file and driver class) as shown in the boxes below the program.

```
#include <iostream>
using namespace std;
class Calculator
{
private:
    int a;
    int b;
public:
    void readNumbers()
    {
        a = 50;
        b = 30;
    }
    void printNumbers()
    {
        cout << "The first number is = " << a << endl;
        cout << "The second number is = " << b << endl;
    }
    int calculateFunction()
    {
        return (a + b);
    }
};

int main()
{
    Calculator num;
    num.readNumbers();
    num.printNumbers();
    cout << "Addition of the two numbers is =" << num.calculateFunction();
    return 0;
}
```

```
//Header file
#ifndef CALCULATOR_H
#define CALCULATOR_H
class Calculator
{
private:
    int a;
    int b;
public:
    void readNumbers();
    void printNumbers();
    int calculateFunction();
};
#endif
```

```
//.cpp file or source file
#include <iostream>
#include "Calculator.h"
using namespace std;

void Calculator::readNumbers()
{
    a = 50;
    b = 30;
}
void Calculator::printNumbers()
{
    cout << "The first number is = " << a << endl;
    cout << "The second number is = " << b << endl;
}
int Calculator::calculateFunction()
{
    return (a + b);
}
```

```
//driver file (containing main function)
#include <iostream>
#include "Calculator.h"
using namespace std;
int main()
{
    Calculator num;
    num.readNumbers();
    num.printNumbers();
    cout << "Addition of the two numbers is = " <<
num.calculateFunction();
    return 0;
}
```

(2-b): Complete the following program by writing the code **(in relevant spaces only)** to define/initialize the static variable (staticVariable). What will be the output of the program if you first initialize the staticVariable with 0, then with 5 and finally without initialization value. (Note: dry run your program for each initialized value, separately)

```
#include <iostream>
using namespace std;
class Test
{
    static int staticVariable;
        //you cannot initialize here    //space for code

public:
    Test()
    {
        // you cannot initialize here    //space for code

        staticVariable++;
    }
    static int getStaticVariable()
    {
        return staticVariable;
    }
};
int Test::staticVariable=0; //space for code
int main()
{
    cout << "The initial value of the staticVariable is : ";
    cout << Test::getStaticVariable() << endl;
    Test objects[5];
    cout << "The updated value of the staticVariable is : ";
    cout << Test::getStaticVariable();
}
```

Output: with initialization 0:

The initial value of the staticVariable is : _____0_____

The updated value of the staticVariable is : _____5_____

Output: with initialization 5:

The initial value of the staticVariable is : _____5_____

The updated value of the staticVariable is : _____10_____

Output: without initialization:

The initial value of the staticVariable is : _____0_____

The updated value of the staticVariable is : _____5_____

(2-c): The output of the following program is 100. Write your code in **two different ways** in the spaces mentioned in the program for a constructor to initialize the data members, where the constructor's parameter and data member have the same name (e.g., both have the name i).

```
#include <iostream>
using namespace std;
class Test
{
    int i;
public:
    Test(int);
    int getValue() const
    {
        return i;
    }
};
//write your code using first way here

Test::Test(int i ){
    this->i=i;
}

//write your code using second way here

Test::Test(int i):i(i){}

int main()
{
    Test object(100);
    cout << object.getValue();
    return 0;
}
```

Question 3 [3+3+9]

(3-a): What will the following program display on screen? Explain the error or bug if there is any.

```
#include <iostream>
using namespace std;
class check {
    float z;
    int x, y;
public:
    check(int x = 0, int y = 1) :x(x + 5), y(y + 2)
    {
        z = x + y + 1;
    }
    void print() {
        cout << " X= " << x
             << " Y = " << y
             << " Z = " << z;
    }
};
int main()
{
    check c;
    c.print();
    return 0;
}
```

Output: X= 5 Y = 3 Z = 2

(3-b): What will the following program display on screen? Explain the error or bug if there is any.

```
#include <iostream>
using namespace std;
class Puzzle
{
    int v1, v2, v3;

    void Init(int x1, int y1)
    {
        v1 = x1;
        v2 = y1;

        int x = x1 + 5;
        int y = y1 + 10;
        v3 = x + y;
    }
    void Display()
    {
        cout << " V1= " << v1
              << " V2= " << v2
              << " V3 = " << v3;
    }
};

int main() {
    Puzzle p;
    p.Init(5, 10);
    p.Display();
    return 0;
}
```

Error as Init() and Display() function are private and cannot be accessed from Main functions

(3-c): Create a class `cTemperature`, which records a temperature. Your program should record temperatures for “n” number of days given by user. For this purpose, your class should have a pointer variable of appropriate type for recording temperature. There are two possible levels for temperature: high and low. Use appropriate OOP concept to incorporate the functionality of level inside the class. Based on temperature the weather report should be printed.

Provide a friend function ***PrintWeather()*** that should be used to print the weather in-terms of temperature. Use dynamic memory allocation for pointers. Use appropriate constructors and destructors for initialization and proper de-allocation of memory for the pointers. Use this pointer where required.

Define ***main()*** function to implement functionalities of `cTemperature` class. Also, print weather report using at least one instance/object of `cTemperature` class. (Note: use the backside of the page if required)

```
class cTemperature
{
    float *temps;
    bool status;
    int size;

public:
    cTemperature(int n=2)
    {
        size = n;
        temps = new float[size];

        cout << "\nEnter " << size << " values of temperature";
        for (int i = 0; i < size; i++)
        {
            cout<< "\nEnter " << i << " values of temperature";
            cin >> temps[i];
        }

        float total=0;

        for (int i = 0; i < size; i++)
        {
            total+= temps[i];
        }

        total = total / size;

        if (total > 20)
            status = true;
        else
            status = false;
    }
};
```

```
    }

    ~cTemperature()
    {
        delete[] temps;
    }
    friend void printWeather(void);

};

void printWeather(void)
{
    cTemperature c1(5);
    if (c1.status == true)
        cout << "\n HIGH Temperature";
    else
        cout << "\n LOW Temperature";
}

int main()
{
    printWeather();
}
```