



School of Computer Science Engineering and Information Systems

Winter Semester 2023-2024

Continuous Assessment Test – II

Programme Name & Branch B.Tech. IT

Course Name & Code: BCSE102L Structured and Object Oriented Programming

Class Number (s): VL2023240503278, VL2023240503275, VL2023240503263

Faculty Name (s) Dr. Chandra Mouliswaran S, Dr. Thanapal P, Dr. Rampriya R S

Exam Duration: 90 Min.

Maximum Marks: 50

General instruction(s): Answer all the Questions

| Q.No. | Questions | Max Marks |
|-------|---|-----------|
| 1. | <p>Write a C program using Structure which calculates the monthly Budget of a person. The program will read the monthly “salary” of the person. Then give following options: Press 1 to calculate gas bill charges Press 2 to calculate electricity bill charges Press 3 to calculate fuelling charges. Press 4 to calculate house rent charges His monthly gas bill charges are 10% of his “salary”, electricity bill charges are 5% of his “salary”. His monthly fuelling charges are 10% of his “salary”, and his house rent charges are 15% of his “salary”. Your program should use that will calculate and display the corresponding charges based on the selection.</p> <p>Answer Key</p> <p>Refer the Logic of the above program</p> <pre>#include<iostream> using namespace std; int main(){ cout<<"Press 1 to calculate gas bill charges\n"; cout<<"Press 2 to calculate electricity bill charges\n"; cout<<"Press 3 to calculate fueling charges.\n"; cout<<"Press 4 to calculate house rent charges\n"; int x; cin>>x; cout<<"Enter the monthly salary\n"; float sal; cin>>sal; if(x==1){ cout<<"Gas bill charges are: "<<0.1 * sal<<endl; } else if(x==2){ cout<<"Electricity bill charges: "<<0.05 * sal<<endl; } else if(x==3){ cout<<"Fueling charges are: "<<0.1 * sal<<endl; } else if(x==4){ cout<<"House rent charges: "<<0.15 * sal<<endl; } }</pre> | 10 |

| | | |
|----|---|----|
| | <p>Mark Distribution: Using Structure and its variable. 4 M Options display and getting input. 2 M Logic using Branching/Looping Statements 4 M</p> | |
| 2. | <p>Create a C++ program with class by name “Employee”. The “Employee” class should have the following data members</p> <ol style="list-style-type: none"> 1. “Employee_Id” 2. “Employee_Name” 3. “Position” 4. “Department” 5. “Basic Pay” 6. “Deductions” (Car Loans, Home Loans, and any other type of loans). <p>The “Employee” class should have the following member functions.</p> <ol style="list-style-type: none"> 1. “Accept()” - It is used to get input for all data members. 2. “Total_Salary()” – calculates the salary includes “HRA” 25%, “DA” 12% & “Vehicle_allowance” 15% of “Basic_Pay”. “Salary” is calculated by the formula “Salary” = “Basic_Pay” + “HRA” + “DA” + “Vehicle_allowance” – “Deductions”. 3. “Display()” - display all the data that has been received as input along with the calculated “Salary”. <p>Create “N” number of “Employee” details with the necessary object for this class.</p> <p>Answer Key Refer the Logic of the above program</p> <pre> #include<iostream> #include<string> using namespace std; class Employee{ private: string Employee_id; string Employee_name; string Position; string Department; double Basic_pay; double Deductions; public: Employee(){} void Accept (){ cin.ignore(); cout<<"Enter employee id: "; getline(cin,Employee_id); cout<<"Enter employee name: "; getline(cin,Employee_name); cout<<"Enter employee Position: "; getline(cin,Position); cout<<"Enter employee Department: "; getline(cin,Department); cout<<"Enter employee Basic_pay: "; cin>>Basic_pay; cout<<"Enter employee Deductions: "; cin>>Deductions; } double Total_salary(){ return Basic_pay + Basic_pay * 0.25 + Basic_pay * 0.12 + Basic_pay * 0.15 - Deductions; } } </pre> | 10 |

```

] void Display(){
-     cout<<"Employee id: "<<Employee_id<<"\n";
-     cout<<"Employee name: "<<Employee_name<<"\n";
-     cout<<"Employee Position: "<<Position<<"\n";
-     cout<<"Employee Department: "<<Department<<"\n";
-     cout<<"Employee Basic_pay: "<<Basic_pay<<"\n";
-     cout<<"Employee Deductions: "<<Deductions<<"\n";
-     cout<<"Employee Salary: "<<Total_salary()<<"\n";
- }
- }
- };
] int main(){
-     int n;
-     cout<<"Enter the number of employees: ";
-     cin>>n;
-     Employee** employees=new Employee*[n];
-     for(int i=0; i<n; i++){
-         employees[i]=new Employee();
-         employees[i]->Accept();
-     }
-     for(int i=0; i<n; i++){
-         employees[i]->Display();
-     }
-     for(int i=0; i<n; i++){
-         delete employees[i];
-     }
-     delete employees;
-     cin>>n;
-     return 0;
- }

```

Mark Distribution:

Using class declare given data member. **2 M**

Using class declare/define given member function Accept(). **1.5 M**

Using class declare/define given member function Total_Salary() with Salary Calculation. **2 M**

Using class declare/define given member function Display(). **1.5 M**

Using array of object for accessing class members. **3 M**

3. Create a class named "Rectangle" with two data members- "length" and "breadth" and a function to calculate the "area" which is "length*breadth". The class has two constructors which are:
- 1 - having no parameter - values of both "length" and "breadth" are assigned zero.
 - 2 - having two numbers as parameters - the two numbers are assigned as "length" and "breadth" respectively.
- Now, create objects of the "Rectangle" class having none and two parameters and print their areas.

Answer Key

Refer the Logic of the above program

10

```

#include <iostream>
using namespace std;

class Rectangle { //define class
public:
    Rectangle() { // default constructor
        length = 0;
        width = 0;
    }

    Rectangle(float l, float w) { //parametrized constructor
        length = l;
        width = w;
    }

    float area() {
        return length * width;
    }

private:
    float length, width;
};

int main() {
    Rectangle r1;
    Rectangle r2(5,7);
    cout<<"Area 1: "<<r1.area()<<endl;
    cout<<"Area 2: "<<r2.area()<<endl;
    return 0;
}

```

Mark Distribution:

Default constructor definition with 0 initialization **2 M**

Parameterised constructor definition. **2 M**

Calculate Area(). **2 M**

Declare Object for default constructor. **1 M**

Declare Object for parameterised constructor. **1 M**

Display Area using default constructor. **1 M**

Display Area using parameterised constructor. **1 M**

- | | | |
|----|---|----|
| 4. | <p>Implement a C++ Program with class named "Travel" and makes private data members, namely "speed" and "distance". Define a member function to assign the private data member's values from the user. Declare a global friend function which calculates the time taken for the Travel. Subsequently, define the same global friend function to find the time of travel not tied to class. Within the global friend function, calculate the time of Travel using the formula "$time = distance / speed$". In main function, create an object for the Travel class. Assign values to the object and call the global friend function to display the time taken for the Travel.</p> <p>Answer Key Refer the Logic of the above program</p> | 10 |
|----|---|----|

```

#include <iostream>
using namespace std;
// Creating a class named Travel.
class Travel {
private:
    // Make private data members, namely speed and distance.
    int speed;
    int distance;
public:
    // A member function to assign the private data member's values from the user.
    void set_values (int a, int b)    {
        speed = a;
        distance = b;
    }
    // A global friend function which calculates the time taken for the Travel.
    friend double findTimeofTravel (Travel); // Friend function
};
// Global Function to find the time of Travel not tied to class.
double findTimeofTravel (Travel t) {
    // Calculate the time of Travel, i.e., distance/speed.
    double time = t.distance / t.speed;
    return time;
}
int main () {
    Travel t; // Create an instance of the Travel class.
    t.set_values(10, 30); // Assign values to the instance t.
    // Call the global friend function to calculate the time taken for the Travel.
    cout << "Time of Travel: " << findTimeofTravel (t) << " hrs" << endl;
    return 0;
}

```

Mark Distribution:

Class Declaration. **1 M**

Declare data members using Private. **1 M**

Define a member function to assign the private data member's values from the user. **1 M**

Declare a global friend function which calculates the time taken for the Travel. **1 M**

Define the same global friend function to find the time of travel not tied to class. Within the global friend function, calculate the time of Travel using the formula " $time = distance / speed$ ". **2 M**

In main function, **1 M**

Create an object for the Travel class. **1 M**

Assign values to the object **1 M**

Call the global friend function to display the time taken for the Travel. **1 M**

5. Design a C++ code for inheritance that considers three classes namely grandfather, father and child. Each one of them has a character data type member. The father is derived from the class grandfather. Similarly the class child is derived from class father. The class grandfather is a base class of the class father. The class father is a base class of the class child. The class child is derived from the class father. The class father is intermediate classes that act as a base class as well as derived class. Display the name of grandfather, father and child.

Answer Key

Refer the Logic of the above program

10

```

#include <iostream>
using namespace std;
class grandfather{
private:
    string name;
public:
    grandfather(string n){
        name=n;
    }
};
class father: public grandfather{
private:
    string name;
public:
    father(string n,string n2): grandfather(n){
        name=n2;
    }
};
class child: public father{
private:
    string c_name;
    string f_name;
    string g_name;
public:
    child(string n,string n2, string n3):father(n,n2){
        g_name=n;
        f_name=n2;
        c_name=n3;
    }
    void display(){
        cout<<"\nName of grandfather: "<<g_name;
        cout<<"\nName of father: "<<f_name;
        cout<<"\nName of child: "<<c_name;
    }
};
int main()
{
    child c("Ramasamy","Rajendran","Shree Rakshn");
    c.display();
    return 0;
}

```

Mark Distribution:

Grandfather class. **2.5 M**

Father class. **2.5 M**

Child class define with multi-level inheritance. **2.5 M**

In main() declare object for derived class and call function. **2.5 M**