



National University of Computer & Emerging Sciences, Karachi
Spring-2021 (School of Computing)
Midterm-I



Course Code: CS-217	Course Name: Object-oriented Programming
Instructor: Dr. Abdul Aziz / Basit Ali / Behraj Khan / Syed Zain Ul Hassan / Talha Khan / NidaMunawar	
Student ID:	Section:
Date: Mar 15, 2021	Time: 11 a.m. - 12 p.m. (60 Minutes)

Instructions:

- Attempt all tasks.
- The paper contain **3** questions on **2** pages.
- Return the paper after the exam.

Max Points: 40

Question 1. Identify the errors with appropriate reasons if any and provide the error free code.

(5 points each)

(Expected time to attempt 05-10 minutes)

a) <pre>#include<iostream> using namespace std; class A { int *x; public: A() { x = new int; } memory_allocate() { x = new int; } memory_free() { delete x; } ~A() { delete x; } }; int main() { A a; a.memory_free(); }</pre>	b) <pre>#include<iostream> using namespace std; class ABC { int x; static int y; public: ABC() { x = 0; y = 0; } ABC(int a, int b) { x = a; y = b; } }; int main() { ABC a; ABC b(1,2); }</pre>
---	---

<p>a)</p> <pre> #include<iostream> using namespace std; class A { int *x; public: A() { x = new int; } memory_allocate() { x = new int; } memory_free() { delete x; x = NULL; } ~A() { if(x!=NULL) { delete x; } } }; int main() { A a; a.memory_free(); } </pre>	<p>b)</p> <pre> #include<iostream> using namespace std; class ABC { int x; static int y; public: ABC() { x = 0; y = 0; } ABC(int a, int b) { x = a; //y = b; set_y(b); } static set_y(int b) { y = b; } }; int ABC::y=0; int main() { ABC a; ABC b(1,2); } </pre>
--	--

Question 2. Short answer questions (answer in a sentence or two):

(3 points each)

(Expected time to attempt 05-10 minutes)

a) When the user-defined copy constructor is required?

(Deep copy) A user-defined copy constructor is generally needed when an object owns pointers or non-shareable references, such as to a file.

b) Is it necessary to have a setter function if we have already defined a parameterize constructor?

Yes, you assign/initialize/set the value using a constructor or setter approach, but after object creation a setter is required when you want to change the value of a private attribute.

c) How can we initialize the constant data members present inside a class? You should add a code snippet to support your answer.

Through a **member initialize list**.

```
class MyClass
{
private:
    constint x;
public:
    MyClass(int a) : x(a){
        //constructor
    }
};
```

d) Constructor is a function and still does not return a value, why?

Because it's **not** called directly by your code, it's called by the memory allocation and object initialization code in the runtime. Its **return value** (if it actually **has** one when compiled down to machine code) is opaque to the user - therefore, you **can't** specify it.

Question 3. You are tasked with developing an application for a microfinance bank which offers loan to customers on easy installments and low interest rate.

(2 Points each)

(Expected time to attempt 30-35 minutes)

In order to issue loans, your application must keep track of customer details including their first name, last name, NIC#, permanent address, current city, annual income and also whether the customer is a tax filer or not. Every customer is going to have a different NIC#, but the NIC# will remain the same for each individual customer.

The bank has several branches across the country and each of these branches are identified by their city, & also contain a branch code and are managed by a branch supervisor. Furthermore, different branch can offer loans to their customers at different interest rates, however, only one interest rate is used by each branch. The interest rate can either be 1.5%, 2.0% or 3.5%.

Perform the following tasks:

- Identify all the entities in the given scenario.
- Create a class diagram / object interaction diagram.
- Create any one class and define attributes and related functions.
- Add a constructor in the class created in part c and initialize their relevant attribute.
- Write relevant C++ statements to declare three branches of the microfinance bank.
- Write a member function named **IssueLoan** that takes customer details and requested loan amount as inputs and confirms loan issuance (via string message) if the loan amount does not exceed the annual income of the said customer and the customer resides in the same city where the branch is located.

- g) Write a function named ***LoanCalculator*** that takes a requested loan amount as input and displays the total payable amount (including interest)
- h) Implement the following behavior in your identified class.
- ```

Main()
{
 Class obj1.
 Class obj2(obj1)
}

```
- i) Provide a mechanism to track the overall loan (total loan amount) issued to all customers combined.

### **Solution:**

#### **Part A)**

- ✓ Customer
- ✓ Branch

#### **Part B)**

| Customer                                                                                                                                           |
|----------------------------------------------------------------------------------------------------------------------------------------------------|
| - firstName: string<br>- lastName : string<br>- NIC : string<br>- address : string<br>- city : string<br>- taxFiler : bool<br>- annualIncome : int |
|                                                                                                                                                    |

#### **Part C to Part I)**

```

class Branch
{
 public:
 // making these variables public for demo purpose (avoiding accessors/mutators)
 string city, branchCode, supervisor;
 const float interest;
 // Part I
 static int totalLoan;

 // Part D
 Branch(string city, string branchCode, string supervisor, float i): interest(i)
 {
 this->city = city;
 this->branchCode = branchCode;
 this->supervisor = supervisor;
 }
 }

```

```

 }

 // Part H
 Branch(const Branch& o): interest(o.interest)
 {
 city = o.city;
 branchCode = o.branchCode;
 supervisor = o.supervisor;
 }

 // Part F
 string IssueLoan(string ci, string fn, string ln, string nic, string ad, int as, bool tf, int amount)
 {
 if((this->city == ci) && (amount <= as))
 {
 totalLoan += amount; // for Part I
 return "Loan request accepted \n";
 }
 else
 return "Loan request rejected \n";
 }

 // Part G
 void LoanCalculator(int amount)
 {
 cout << "Total payable amount: " << amount * interest << endl;
 }
};

int Branch::totalLoan = 0;

class Customer
{
public:
 // making these variables public for demo purpose (avoiding accessors/mutators)
 string city, firstName, lastName, address;
 const string NIC;
 int annualSalary;
 bool taxFiler;

 // Part D
 Customer(string city, string firstName, string lastName, string nic, string address, int
annualSalary, bool taxFiler): NIC(nic)
 {
 this->city = city;
 this->firstName = firstName;
 this->lastName = lastName;

 this->address = address;
 this->annualSalary = annualSalary;
 this->taxFiler = taxFiler;
 }
}

```

```

};

int main()
{
 // Part E
 Branch branch1("Karachi", "K010", "Mahmood Ahmed", 2.0f);
 Branch branch2("Lahore", "L020", "Zakir Hussain", 3.5f);
 Branch branch3("Faisalabad", "F030", "Arsalan Khan", 1.5f);

 // Part H
 Branch branch4(branch1);

 // Customer c1("Karachi", "Amir", "Ali", "42501-1234567-8", "Khayaban-e-Sahar", 1200000,
true);
 cout << branch1.IssueLoan("Karachi", "Amir", "Ali", "42501-1234567-8", "Khayaban-e-Sahar",
1200000, true, 1100000);
 branch1.LoanCalculator(1100000);
}

```