

## Object Oriented Programming (CS1004)

Course Instructor(s):

Ms. Momina Behzad

Section(s): SE-A

*Solution*

Roll No

Course Section

## Sessional-I Exam

Total Time (Hrs): 1

50

Total Marks:

3

Total Questions:

Date: Sep 23, 2025

Do not write below this line.

Attempt all the questions.

[CLO 1: Demonstrate the basic concepts of OOP]

[2.5\*4=10 marks]

Q1: Complete the following code snippets with correct syntax:

Part A:

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

```
class Book {
    string title
    int pages
public:
    Book(string t, int p)
    ~Book()
    void print()
}

Book :: Book(string t, int p) {
    title = t
    pages = p
    cout << "Book created: " << title <<
    Pages: " << pages << endl;
}

Book :: ~Book() {
    cout << "Book destroyed: " << title <<
    endl;
}

void Book :: print() {
    cout << "Title: " << title << " Pages: "
    << pages << endl;
}

int main() {
    Book b1("C++ Basics", 300)
    b1.print()

    Book* b2 = new Book("OOP Concepts", 500)
    b2 -> print()
```

Part B:

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

```
class Car {
    string brand
    int speed
public:
    Car(string b, int s)
    ~Car()
    void accelerate(int val)
}

Car :: Car(string b, int s) {
    brand = b
    speed = s
    cout << "Car created: " << brand <<
    Speed: " << speed << endl;
}

Car :: ~Car() {
    cout << "Car destroyed: " << brand <<
    endl;
}

void Car :: accelerate(int val) {
    speed = speed + val
    cout << brand << " new speed: " <<
    speed << endl;
}
```

```
int main() {
    Car c1("Toyota", 120)
    c1 -> accelerate(30)

    Car* c2 = new Car("Ferrari", 200)
    c2 -> accelerate(50)
```

# National University of Computer and Emerging Sciences

## Islamabad Campus

```
    delete b2;
}
return 0;
}
```

### Part C:

```
#include <iostream>
using namespace std;
class MyClass {
    int x;
public:
    MyClass(int val)
        ~MyClass()
    void foo()
}
MyClass::MyClass(int val) {
    << "Constructor called with value "
    << val << endl;
    x = val;
}
MyClass::~MyClass() {
    cout << "Destructor called for " << x
    << endl;
}
void MyClass::foo() {
    cout << "Value is " << x << endl;
}
int main() {
    MyClass obj(10);
    obj.foo();
    MyClass* ptr = new MyClass(20);
    ptr->foo();
    delete ptr;
    return 0;
}
```

```
    delete c2;
}
return 0;
}
```

### Part D:

```
#include <iostream>
using namespace std;
class Player {
    string name;
    int score;
public:
    Player(string n, int s)
        ~Player()
    void updateScore(int add)
}
Player::Player(string n, int s) {
    name = n;
    score = s;
    cout << "Player joined: " << name << endl;
    cout << "Score: " << score << endl;
}
Player::~Player() {
    cout << "Player left: " << name << endl;
}
void Player::updateScore(int add) {
    score = score + add;
    cout << "Updated Score of " << name
    << ":" << score << endl;
}
int main() {
    Player p1("Ali", 50);
    p1.updateScore(25);
    Player* p2 = new Player("Sara", 70);
    p2->updateScore(40);
    delete p2;
    return 0;
}
```

# National University of Computer and Emerging Sciences

## Islamabad Campus

[CLO 2: Apply OOP concepts to Solve the problem]

Q2: Apply the concepts of class and objects to complete the following problem. (Attempt this question on answer sheet) [15 marks]

Part A: Write header and cpp file to execute the following main procedure. (10 Marks)

```
#include <iostream>
#include "Gadget.h"
using namespace std;

int main() {
    Gadget g1(101, "Smart Watch", 15000.50);
    g1.display();

    g1.updatePrice(17999.99);
    cout << "Updated ID: " << g1.getId() << endl;
    g1.display();

    return 0;
}
```

Part B: Write header and cpp file for the following class. (5 Marks)

Cat
+ name
- age
+ weight
+ Cat()
+ ~Cat()
+ meow()
+ eat()
+ sleep()

Q2

Date: \_\_\_\_\_

## Header

```
class Gadget
{
private:
    string name;
    int ID;
    double price;
public:
    void updatePrice (double p);
    void display();
    Gadget ();
    Gadget (int id, string name, double price);
    int getID();
```

## Source file

```
#include "Gadget.h"
Gadget :: Gadget()
{
    id = 0;
    name = "null";
    price = 0;
}
Gadget :: Gadget (int id, string name, double price)
{
    id = id;
    name = name;
    price = price;
}
void Gadget :: display ()
{
    cout << "ID" << id <<
    "name" << name <<
    "price" << price;
}
int Gadget :: getID()
{
    return id;
}
void Gadget :: updatePrice (double p)
{
    price = p;
```

## Q2 (Part B)

Date: \_\_\_\_\_

Header

Class Cat

{  
Private:

int age;

Public:

string name;  
~~double~~ weight;

Cat();

~Cat();

void meow();

void eat();

void sleep();

} ;

Source

#include "Cat.h"

Cat::Cat()

{

age = 0;

name = "",

weight = 0;

{

~Cat::~Cat()

{

cout << "Cat "

"Destroy";

{

void Cat::meow()

{

cout << "meow";

{

void Cat::eat()

{

cout << "cat is

eating";

{

void Cat::sleep()

{

cout << "cat is  
sleeping";

{

# National University of Computer and Emerging Sciences

## Islamabad Campus

### [CLO 2: Apply OOP concepts to Solve the problem]

**Q3:** Trace the execution of the following code segment. If the code is syntactically correct, write its output. If the code contains errors, rewrite the corrected version of the code and then provide the output.

**Part A: (3\*5=15)**

Code	Corrected Version or Output
<b>Part A:</b> <pre>void foo(int *arr, int n) {     if (n &lt;= 0) return;     cout &lt;&lt; *arr &lt;&lt; " ";     if (n % 2 == 0) foo(arr + 1, n - 2); }  int main() {     int arr[] = {1, 2, 3, 4, 5};     foo(arr, 5);     return 0; }</pre>	<b>1</b>
<b>Part B:</b> <pre>void foo(int *p, int *end) {     if (*p &gt;= *end) return;     cout &lt;&lt; *p &lt;&lt; " ";     foo(p + *p % 3, end - 1); }  int main() {     int arr[] = {4, 2, 5, 3, 1};     foo(arr, arr + 5);     return 0; }</pre>	<b>4      2</b>
<b>Part C:</b> <pre>void foo(int *a, int *b, int depth) {     if (depth == 0) return;     int temp = *a;     *a = *b;     *b = temp;     foo(a, b - 1, depth - 1); }  int main() {     int arr[] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};     foo(arr, arr + 4, 3);     for (int i = 0; i &lt; 10; i++) cout &lt;&lt; arr[i] &lt;&lt; " ";     return 0; }</pre>	<b>30 20 40 50 10 60 70 80 90 100</b>
<b>Part D:</b> <pre>void foo(int *start, int *end) {     if (*start &gt;= *end) return;     cout &lt;&lt; *start &lt;&lt; endl;     cout &lt;&lt; *end &lt;&lt; " " &lt;&lt; endl;     cout &lt;&lt; *start + *end &lt;&lt; " ";     foo(start + 1, end - 1);     cout &lt;&lt; *start - *end &lt;&lt; " "; }</pre>	<b>1 6 7 2 5 7 3 4 7 -1 -3 -5</b>

National University of Computer and Emerging Sciences  
Islamabad Campus

```
int main() {
    int arr[] = {1, 2, 3, 4, 5, 6};
    foo(arr, arr + 5);
    return 0;
}
```

Part E:

```
void fun1(int *p, int n) {
    if (n == 0) return;
    fun1(p + 1, n - 1);
    cout << *p << " ";
}

int main() {
    int arr[] = {5, 10, 15};
    fun1(arr, 3);
    return 0;
}
```

15      10      5

Part B (2\*5=10)

Code	Corrected Version or Output
<p>Part A:</p> <pre>#include &lt;iostream&gt; using namespace std;  class Device {     int id;     string name;     double price;  public:     Device() {         cout &lt;&lt; "X1 triggered" &lt;&lt; endl;         id = 0;         name = "N/A";         price = 0.0;         cout &lt;&lt; "Init done with defaults" &lt;&lt; endl;     }      Device(int i, string n, double p) {         cout &lt;&lt; "X2 triggered" &lt;&lt; endl;         id = i;         name = n;         price = p;         cout &lt;&lt; "Init done with custom values" &lt;&lt; endl;         cout &lt;&lt; "Values -&gt; " &lt;&lt; id &lt;&lt; "   " &lt;&lt; name &lt;&lt; "   "         " &lt;&lt; price &lt;&lt; endl;     }      void foo1() {         cout &lt;&lt; "M1 called" &lt;&lt; endl;         cout &lt;&lt; id &lt;&lt; " - " &lt;&lt; name &lt;&lt; " - " &lt;&lt; price &lt;&lt;         endl;     } }</pre>	<p>&gt;&gt;&gt; START &lt;&lt;&lt;</p> <p>X1 Triggered</p> <p>Init done with defaults</p> <p>M1 called</p> <p>O - NA - O</p> <p>X2 triggered</p> <p>Init done with custom values.</p> <p>values → 101   smartphone   699.99</p> <p>M1 called</p> <p>101 - smartphone - 699.99</p> <p>M2 called</p> <p>Before : 699.99</p> <p>After : 799.99</p> <p>M1 called</p> <p>101 - smart phone - 799.99</p> <p>X2 triggered</p>

National University of Computer and Emerging Sciences  
Islamabad Campus

```

void foo2(double newPrice) {
    cout << "M2 called" << endl;
    cout << "Before: " << price << endl;
    price = newPrice;
    cout << "After: " << price << endl;
}

~Device() {
    cout << "Z1 released -> " << name << endl;
}

int main() {
    cout << ">>> START <<<" << endl;

    Device d1;
    d1.foo1();

    Device d2(101, "Smartphone", 699.99);
    d2.foo1();

    d2.foo2(799.99);
    d2.foo1();

    Device* d3 = new Device(202, "Laptop", 1299.50);
    d3->foo1();

    d3->foo2(1499.75);
    d3->foo1();

    delete d3;

    cout << ">>> END <<<" << endl;
    return 0;
}

```

**Part B:**

```

#include <iostream>
using namespace std;

class BankAccount {
    int* transactions;
    int size;

public:
    BankAccount() {
        cout << "X1 init" << endl;
        size = 5;
        transactions = new int[size];
        for (int i = 0; i < size; i++) {
            *(transactions + i) = (i + 1) * 100;
            cout << "[" << i << "]=" << *(transactions + i)
            << " ";
        }
        cout << endl;
    }
}

```

Unit done with custom values  
values → D02 | Laptop | 1299.5  
M1 called  
D02 - Laptop - 1299.5  
M2 called  
20223  
Before : 1299.5  
After : 1499.75  
M1 called  
D02 - Laptop - 1499.75  
Z1 released → laptop  
>> End <<  
Z1 released → smartphone  
Z1 released → N/A.

>>> START <<<  
X1 INIT  
[0]=100 [1]=200 [2]=300  
[3]=400 [4]=500  
M1 dump: 100 200 300 400 500  
M2 exec 25  
M1 dump: 125 225 325 425 525  
M3 flip  
M1 dump: 525 425 325 225 125  
X2 init

National University of Computer and Emerging Sciences  
Islamabad Campus

```

BankAccount(int n) {
    cout << "X2 init" << endl;
    size = n;
    transactions = new int[size];
    for (int i = 0; i < size; i++) {
        *(transactions + i) = (i + 1) * 50;
        cout << "[" << i << "]=" << *(transactions + i)
        << " ";
    }
    cout << endl;
}

void foo1() {
    cout << "M1 dump: ";
    for (int i = 0; i < size; i++) {
        cout << *(transactions + i) << " ";
    }
    cout << endl;
}

void foo2(int bonus) {
    cout << "M2 exec " << bonus << endl;
    for (int i = 0; i < size; i++) {
        *(transactions + i) += bonus;
    }
}

void foo3() {
    cout << "M3 flip" << endl;
    int* left = transactions;
    int* right = transactions + size - 1;
    while (left < right) {
        int temp = *left;
        *left = *right;
        *right = temp;
        left++;
        right--;
    }
}

~BankAccount() {
    cout << "Z1 release" << endl;
    cout << "Final: ";
    for (int i = 0; i < size; i++) {
        cout << *(transactions + i) << " ";
    }
    cout << endl;
    delete[] transactions;
}

int main() {
    cout << ">> START <<<" << endl;
    BankAccount al;
    al.foo1();
}

```

$[0] = 50 \quad [1] = 100 \quad [2] = 150$   
 $[3] = 200 \quad [4] = 250 \quad [5] = 300$

M1 dump: 50 100 150 200  
250 300

M2 exec 10

M3 ~~flip~~ flip

M1 dump: 310 260 210  
160 110 60

Z1 release

Final: 310 260 210 160  
110 60

>>> End <<<

Final 300 250 200 150

Z1 release

Final: 525 425 325  
225 125

National University of Computer and Emerging Sciences  
Islamabad Campus

```
a1.foo2(25);  
a1.foo1();  
a1.foo3();  
a1.foo1();  
  
BankAccount a2(4);  
a2.foo1();  
a2.foo2(100);  
a2.foo3();  
a2.foo1();  
  
BankAccount* a3 = new BankAccount(6);  
a3->foo1();  
a3->foo2(10);  
a3->foo3();  
a3->foo1();  
  
delete a3;  
  
cout << ">>> END <<<" << endl;  
return 0;
```