National University of Computer and Emerging Sciences
Lahore Campus

# Object Oriented Programming (CS1004)

Date: Feb 27 2025

**Course Instructor(s)**

SK, SI, UA, UN, NA, AK, UA, RD

# Sessional-I Exam

**Total Time (Hrs): 1**
**Total Marks: 20**
**Total Questions: 3**

____________ Roll No

____________ Section

________________________ Student Signature

**Do not write below this line**

**Attempt all the questions on question paper. You can use rough sheet but do not attach it with the question paper. Don not use red ink or pencil to solve the exam.**

***CLO #1: Demonstrate basic concepts of OOP***

**Q1:** Write a C++ function **Word_Positions** that takes a character array ***str*** as input and processes the text to:

1. Count the number of unique words in the text.
2. For all unique words, store the positions (word position) of its occurrences in a dynamic two-dimensional array such that size of each row is number of occurrences+1. The additional memory will be allocated to store delimeter -1 to represent the end of positions.

For the sake of simplicity, you can assume that all characters in str are in lower case. Also all the words are separated by white space and there are no punctuation marks.

**Sample Input:** "hello world this is a sample text hello again world"
**Sample Solution:**

| | | | |
|---|---|---|---|
| hello | **0** | 7 | **-1** |
| world | **1** | **9** | **-1** |
| this | **2** | **-1** | |
| is | **3** | **-1** | |
| a | **4** | **-1** | |
| sample | **5** | **-1** | |
| text | **6** | **-1** | |
| again | **8** | **-1** | |

In the example above, word "hello" is present at word position 0 and 7 whereas word "sample" is present at position 5.

The function should return the number of unique words and a dynamic two-dimensional array of integers where each row represents occurrence positions of a unique word (sample two dimensional array is shown bold in the example above. [10 marks]

*CLO #1: Demonstrate basic concepts of OOP*

**Q2: Find errors in the following code segments and suggest corrections** [3 +1+2marks]

**(a)**

```
int* retrievePointer() {
   int tempVar = 10;
   return &tempVar;
}
void demonstrateArray() {
   int array[5] = { 1, 2, 3, 4, 5 };
   for (int i = 0; i < 5; array++){
      cout << *array<<" ";
   }
}
int main() {
   int* data = new int[10];
   int* dan = retrievePointer();
   cout<<"Pointer Value: " <<*dan;
   demonstrateArray();
   delete data;
   return 0;
}
```

**Error + correction**

**(b) For all other sections**

```
int main() {
   unique_ptr<int> ptr(new int(10));
   unique_ptr<int> ptr2(new int(20));
   ptr2 = ptr;
   cout << "Value: " << *ptr << endl;
   ptr.reset(new int(30));
   cout << "New Value: " <<*ptr;
   return 0;
}
```

**(b) For BDS-2B, BCS-2E, BCS2-F, BCS-2G**

```
#include <iostream>
#include <memory>
using namespace std;
int main() {
   unique_ptr<int> ptr1, ptr2;
   ptr1 = make_unique<int>(100);
   cout << "Value of ptr1: " << *ptr1 << "\n";
   ptr2 = move(ptr1);
   cout << "Value of ptr1: " << *ptr1 << "\n";
   return 0;
}
```

**Error + correction (for part b)**

**(c)**

```
void setPointer(int*& ptr) {
   int temp = 30;
   ptr = &temp;
}
void modifyPointer(int* ptr) {
   *ptr = 50;
}
int main() {
   int val = 10;
   int* ptr = &val;
   cout<<"Before setPointer: "<<*ptr;
   setPointer(ptr);
   cout<<"After setPointer: " <<*ptr;
   modifyPointer(ptr);
   cout<<"After modifyPointer: " <<*ptr;
   return 0;
}
```

**Error + correction**

***CLO 1: Demonstrate basic concepts of OOP***

**Q3:** Give output of the following code segments. [2+2 marks]

```
int arr[] = {100, 200, 300, 400, 500};
   int *ptr1 = arr;
   int *ptr2 = arr + 4;
   cout << "Output 1: " << *(ptr2 - 1) + *(ptr1 + 2) << endl;
   cout << "Output 2: " << *(ptr2) - *(ptr1 + 3) << endl;
   cout << "Output 3: " << *(ptr2) * (*(ptr1 + 1)) << endl;
   cout << "Output 4: " << *(ptr2) / *(ptr1+1) << endl;
   cout << "Output 5: " << (*(ptr2 - 2)) % *(ptr1 + 4) << endl;
```

**Output:**

```
class Circle {
private:
   double radius;
   double centerX;
   double centerY;

public:
   Circle() {
      cout << "Default Constructor: " << endl;
      radius = 0; centerX = 0; centerY = 0;
   }

   void SetCircle(double r, double x, double y) {
      cout << "Set Function: " << endl;
      radius = r; centerX = x; centerY = y;
   }
   void printArea() {
      double area = 3.14159 * radius * radius;
      cout << "Area of the circle: " << area << endl;
   }
};
int main() {
   Circle c1, c2;
   c2.SetCircle(5, 2, 3);
   c1.printArea();
   c2.printArea();
   return 0;
}
```