



University of Central Punjab

Faculty of Information Technology

Mid-Term Exam Fall - 2021

Object Oriented Programming– Lab

Instructions for Invigilators:

1. Students will have total 90 minutes to finish the whole exam. It is up to the students to manage their time.

Instructions for Students:

1. Please create file with appropriate name
2. Submit only **.h** and **.cpp** files with output screenshot on [portal](#).
3. Late submissions will **NOT** be considered
4. Create as many classes and functions as required. Remember one function for one functionality.
5. Take care, plagiarism will not be tolerated at any case.
6. No .rar files are accepted
7. The paper is close book and close notes. No cheat sheet allowed.
8. Use meaningful variable names, take care of naming conventions and indentation. **5 Marks will be deducted for each thing if not followed.**
9. Everything that could be constant, should be constant.



University of Central Punjab

Faculty of Information Technology

Question 1 – 60 Marks

(Constructors=25%, Functions=25%, Operators=40%, Others=10%)

A main() function is given below. Based on this function, your task is to write complete code **required** to run the main() function successfully and produce the exact output given. **You are not allowed to return any memory handler in your code.**

For your ease, a file containing main() function is uploaded on the portal.

Remember:

1. You cannot change anything in the main() function and should produce the exact output given.
2. If your code will not be able to compile, you will not be awarded more than 50% marks even if you have written the whole code.
3. Submit the screenshot of your output as well. It carries marks.

```
int main()
{
    IntegerList list1;           //making empty default IntegerList of 100 size
    int number = 12;
    int i = 1;
    while (i <= number)
    {
        list1.insert(i); //inserting elements into list1 by incrementing its indexes like 0
                          //1,2,3,4 and so on. For eg. 1st time insertion will insert at index 0,
                          //then index 1 and so on
        i++;
    }
    cout << "list1\n";
    print(list1);
    cout << "The number of elements in list1 = " << list1.getCount() << endl;

    IntegerList list2(list1, 2, 9); //should copy the elements of list1 from position 2 to 9
    cout << "list2 is a copy of list1 from position 2 to 9\n";
    print(list2);
    cout << "The number of elements in list2 = " << list2.getCount() << endl;

    IntegerList list3;
    cout << "list3 is a constant IntegerList\n";
    cout << "The number of elements in list3 = " << list3.getCount() << endl;

    list1.clear(); //should set the values of list1 to zeros and its count to zero
    cout << "After clearing list1\n";
    print(list1);

    IntegerList list4(10); //should make an empty IntegerList of size 10(passed as a parameter)
    cout << "list4 is an empty IntegerList of size 10\n";
    print(list4);

    list4.insert(55); //inserting elements into list4
    list4.insert(56); //inserting elements into list4
    list4.insert(57); //inserting elements into list4
    list4.insert(58); //inserting elements into list4
```



University of Central Punjab

Faculty of Information Technology

```
cout << "After inserting elements in list4\n";
print(list4);

IntegerList list5;
list5 = list2 + list4; //list5 should have both the elements of list2 and list4
cout << "list5 contains both the elements of list2 and list4\n";
print(list5);
list5[0] = list5[0] + 1; //it should give compile time error if list5 is constant
                        //For eg: const IntegerList list5
cout << "After adding 1 to the element at 0 index of list5\n";
print(list5);

cout << "print(list5++)\n";
print(list5++);        //post-increment

cout << "print(++list5)\n";
print(++list5);        //pre-increment
cout << "print(--list5)\n";
print(--list5);        //pre-decrement

cout << "print(list5--)\n";
print(list5--);        //post-decrement

cout << "list6 is a copy of list5\n";
IntegerList list6 = list5; //list6 is a copy of list5
print(list6);

cout << "print(list5++)\n";
print(list5++);

cout << "Comparing list5 and list6 are equal or not?\n";
if (list6 == list5)      // to compare all the elements of IntegerLists
{
    cout << "As you can see Both the IntegerLists list5 and list6 are equal\n";
    cout << "list5: ";
    print(list5);
    cout << "list6: ";
    print(list6);
}
else
{
    cout << "As you can see Both the IntegerLists list5 and list6 are not equal\n";
    cout << "list5: ";
    print(list5);
    cout << "list6: ";
    print(list6);
}

cout << "list4 after assignment: ";
(list4 = list5) = list6;
print(list4);

return 0;
}
```



University of Central Punjab

Faculty of Information Technology

Output:

```
C# Microsoft Visual Studio Debug Console

1. list1
2. 1 2 3 4 5 6 7 8 9 10 11 12
3. The number of elements in list1 = 12
4. list2 is a copy of list1 from position 2 to 9
5. 2 3 4 5 6 7 8 9
6. The number of elements in list2 = 8
7. list3 is a constant IntegerList
8. The number of elements in list3 = 0
9. After clearing list1
10. List is empty.
11. list4 is an empty IntegerList of size 10
12. List is empty.
13. After inserting elements in list4
14. 55 56 57 58
15. list5 contains both the elements of list2 and list4
16. 2 3 4 5 6 7 8 9 55 56 57 58
17. After adding 1 to the elmenent at 0 index of list5
18. 3 3 4 5 6 7 8 9 55 56 57 58
19. print(list5++);
20. 3 3 4 5 6 7 8 9 55 56 57 58
21. print(++list5)
22. 5 5 6 7 8 9 10 11 57 58 59 60
23. print(--list5)
24. 4 4 5 6 7 8 9 10 56 57 58 59
25. print(list5--)
26. 4 4 5 6 7 8 9 10 56 57 58 59
27. list6 is a copy of list5
28. 3 3 4 5 6 7 8 9 55 56 57 58
29. print(list5++)
30. 3 3 4 5 6 7 8 9 55 56 57 58
31. Comparing list5 and list6 are equal or not?
32. As you can see Both the IntegerLists list5 and list6 are not equal
33. list5: 4 4 5 6 7 8 9 10 56 57 58 59
34. list6: 3 3 4 5 6 7 8 9 55 56 57 58
35. list4 after assignment: 3 3 4 5 6 7 8 9 55 56 57 58

C:\Users\mbilal.ishfaq\source\repos\IntegerList\Debug\IntegerList.
Press any key to close this window . . .
```