|  |
| --- |
| Serial No: |
| **1st Mid Term Exam**  **Subjective** |
| **Total Time: 50 Minutes** |
| **Total Marks: 40** |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Signature of Invigilator |

|  |
| --- |
| **CS2001 Data Structures** |
| Monday, September 26, 2022 |
| **Course Instructor** |
| Dr. Hashim Yasin  Dr. Anwer Shah  Mr. Muhammad Usman Joyia  Mr. Muhammad Yousuf |

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ . .  Roll No Section Signature |

## DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

**Instructions:**

1. Verify at the start of the exam that you have a total of four (04) questions printed on six (06) pages including this title page.
2. Attempt all questions on the question-book and in the given order.
3. The exam is closed books, closed notes. Please see that the area in your threshold is free of any material classified as ‘useful in the paper’ or else there may a charge of cheating.
4. Read the questions carefully for clarity of context and understanding of meaning and make assumptions wherever required, for neither the invigilator will address your queries, nor the teacher/examiner will come to the examination hall for any assistance.
5. Fit in all your answers in the provided space. You may use extra space on the last page if required. If you do so, clearly mark question/part number on that page to avoid confusion.
6. Use only your own stationery and calculator. If you do not have your own calculator, use manual calculations.
7. Use only permanent ink-pens. Only the questions attempted with permanent ink-pens will be considered. Any part of paper done in lead pencil cannot be claimed for checking/rechecking.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Q-1 | Q-2 | Q-3 | Q-4 | Q-5 | Total |
| **Total**  **Marks** | **10** | **10** | **10** | **10** | **10** | **50** |
| **Marks Obtained** |  |  |  |  |  |  |

**Vetted By: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Vetter Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **University Answer Sheet Required:** | **No** |  |  |  |  | **Yes** |  |

**Question No. 2 Singly & Doubly Linked List [10 Marks]**Consider that **Doubly Link List** (DLL) of some nodes exists in memory. There is a ‘**Current’** pointer, which points to any random node in DLL. Write a C++ code that tells whether the ‘**Current’** pointer is **nearer** to the **End** of **DLL** or **nearer** to the **Head** of **DLL** or having same **Distance** from both sides.

(Write the specific C++ code, no need of #include, main etc.)

|  |
| --- |
| node\*temp = current;      int forwCount=0, backwCount=0;        //count nodes towards End      while(temp!=NULL)      {          forwCount++;          temp=temp->next;      }        //count nodes towards Head      temp=current;      while(temp!=NULL)      {          backwCount++;          temp=temp->previous;      }        //Comparing both the counts for final answer      if(forwCount == backwCount)          cout<<"Current pointer has the same distance from both ends."<<endl;      else if(forwCount > backwCount)          cout<<"Current pointer is nearer to the End of DLL."<<endl;      else          cout<<"Current pointer is nearer to the Head of DLL."<<endl; |

**Question No. 3 Circular Linked List [3+7 = 10 Marks]**Suppose you’re a software developer in the Spotify Application where you are asked to implement the music player with the playlist feature. The playlist functionality is required with the following rules:

1. User can add as many songs as it likes.
2. When user play the playlist, the playlist songs will play recurrently.
3. User can remove any he wants to remove from playlist.

This problem can be dealt with using the linked list data structure. However, to keep the problem a bit simple, songs can be represented as an object of Song ADT implemented already. You’re required to answer the following questions:

1. Design linkedlist ADT required to implement the playlist

|  |
| --- |
|  |

1. Write a piece to code to provide all the songs that user has added in the playlist

|  |
| --- |
|  |

**Question No. 4 Stack [5+5=10 Marks]**Evaluate the following postfix and prefix expressions. **Note**: Write your final answer in the box given as under to get the full marks. You can perform all the steps involved in the process as **rough** work (if it is necessary) on the other side of the paper but it contains **NO** marks.

|  |
| --- |
|  |

|  |
| --- |
|  |

**Question No. 5 Stacks Application [5+5=10 Marks]**

Convert the following infix expression to prefix expression and postfix expression using stack method.

A – (B /C \* (D \* E ^ F) / G) \* H

|  |
| --- |
|  |