

```

/*
 * advanced programming
 * Alireza Akhavan Pour
 * akhavan@alirezaWeb.com
 * date: 1395/02/25
 */
int main()
{
    cout<< "In the name of God";
    Lecture_23 ();
    return 0;
}

string Lecture_23()
{
    cout << "Linked list" <<endl;
    return ":";
}

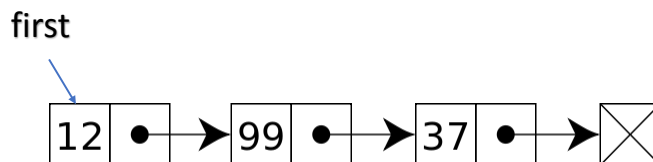
```



1

لیست پیوندی غیر شیء گرا

- در این مثال لیست پیوندی ساده به صورت غیر شیء گرا پیاده سازی می شود. عناصر این لیست اعداد صحیح هستند.

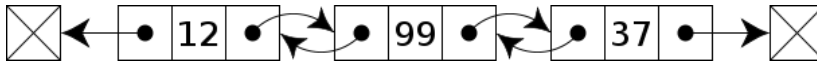


01-NonOOSinglyLinkedList.cpp

2

لیست پیوندی دو طرفه غیر شیء گرا

- در این مثال لیست پیوندی دو طرفه به صورت غیر شیء گرا پیاده سازی می شود. عناصر این لیست اعداد صحیح هستند.



02-NonOODoublyLinkedList.cpp

3

لیست های پیوندی پیاده سازی در قالب کلاس

- در این مثال لیست پیوندی دو طرفه به صورت یک کلاس پیاده سازی می شود. با وجود این که هر اشاره گر به Node را می توان به عنوان عنصر اول یک لیست پیوندی در نظر گرفت، مخفی سازی جزئیات پیاده سازی انگیزه ی ما برای بسته بندی لیست پیوندی در یک کلاس مجزا است.

03-BasicLinkedList.cpp

4

مشكلات

```

int main() {
    List l;

    l.push_back(86);
    l.push_front(43);
    l.push_front(12);

    int sum = 0;
    for (Node* p = l.head(); p != NULL; p = p->next)
        sum += p->data;
    cout << "sum: " << sum << endl;

    l.head()->next = NULL;

    List l2;
    l2.push_back(1);
    l2.push_back(2);
    l2.push_back(3);

    system("pause");
}

```



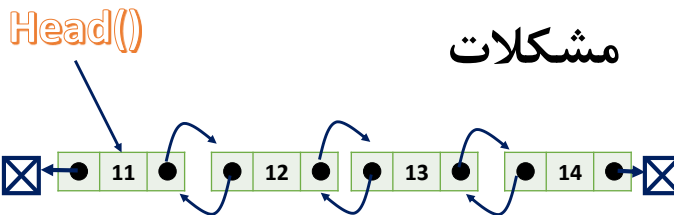
5

مشكلات

```

List l;
l.push_back(11);
l.push_back(12);
l.push_back(13);
l.push_back(14);

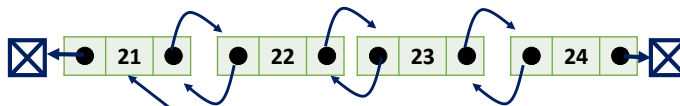
```



```

List l2;
l2.push_back(21);
l2.push_back(22);
l2.push_back(23);
l2.push_back(24);

```



```

Node *p=l.head();
p=p->next;
p=p->next;

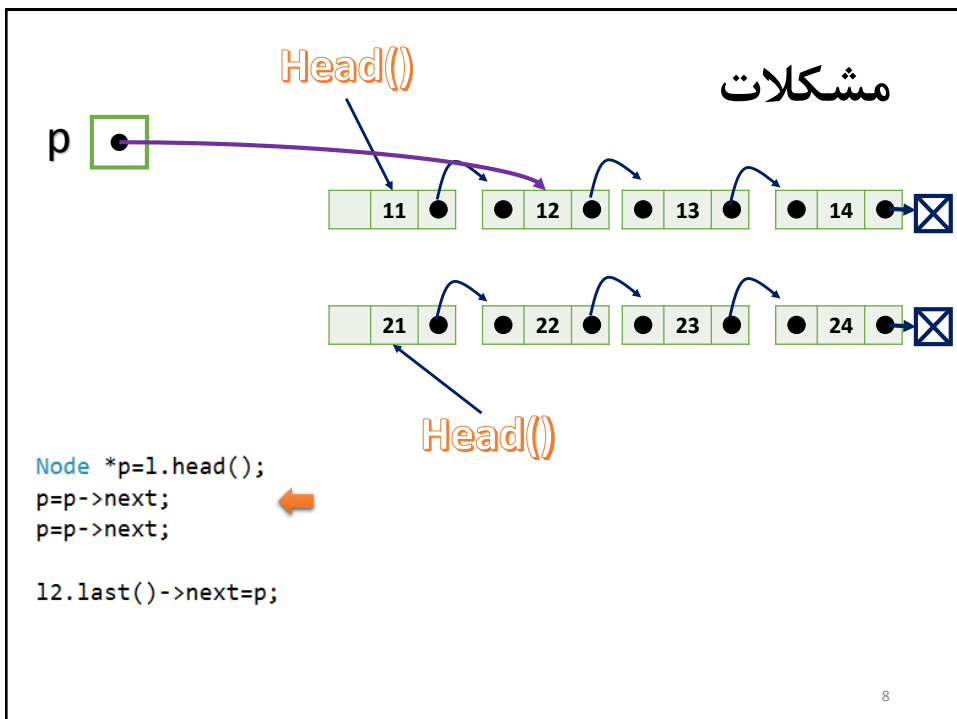
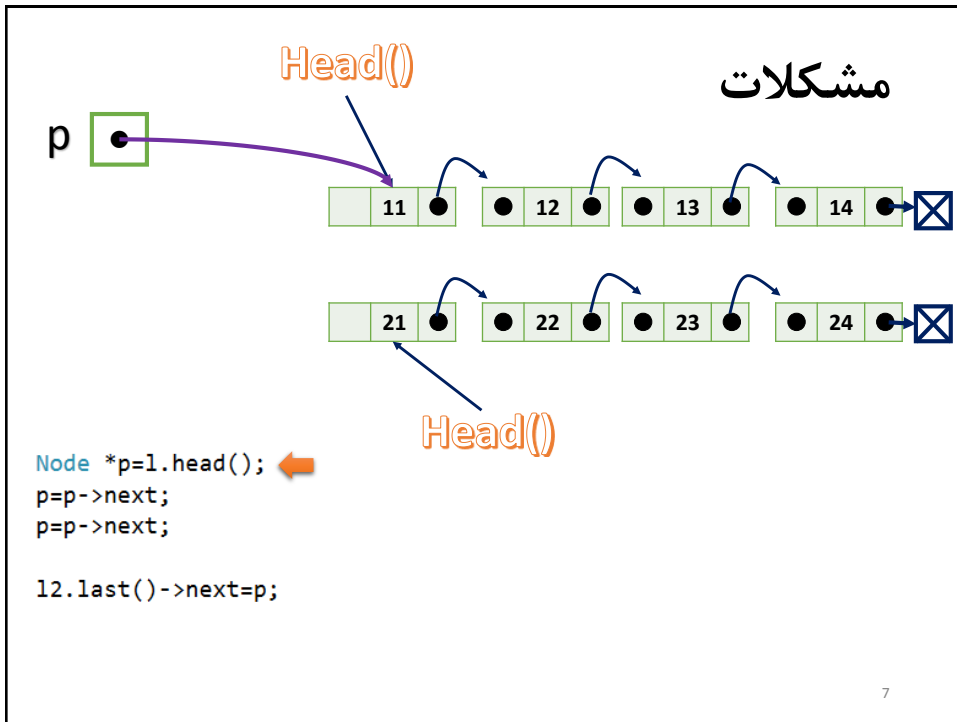
l2.last()->next=p;

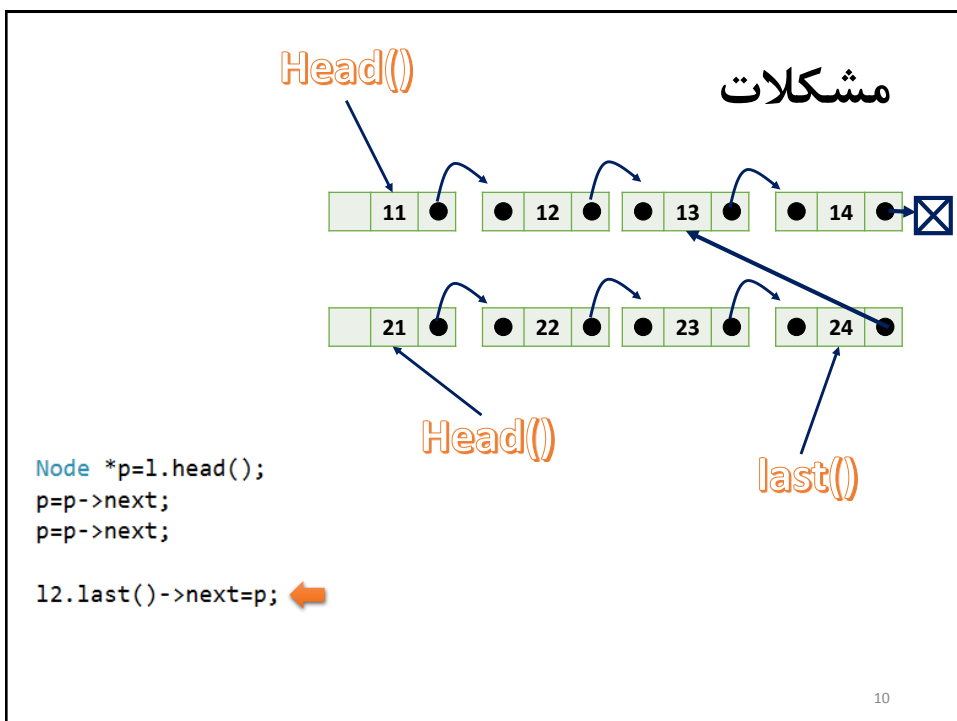
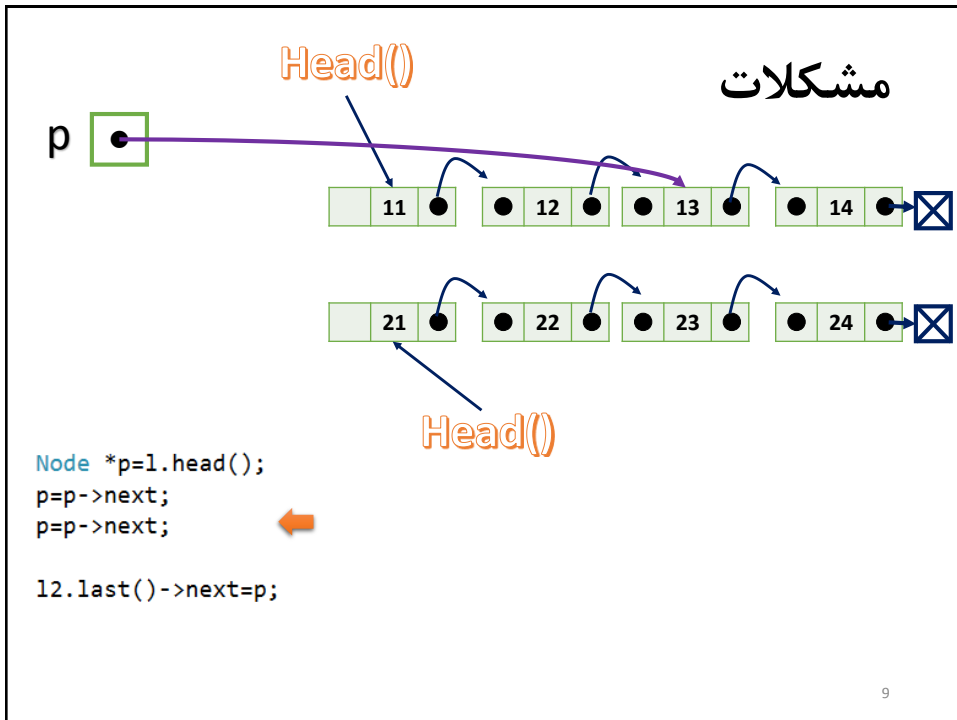
```



04-BasicLinkedListRTErr.cpp

6





منابع

- https://en.wikipedia.org/wiki/Linked_list
- http://ramtung.ir/apnotes/html/14_LinkedLists.html