Started on	Sunday, 9 March 2025, 3:01 PM
State	Finished
Completed on	Sunday, 9 March 2025, 3:04 PM
Time taken	2 mins 48 secs
Marks	4.00/5.00
Grade	**80.00** out of 100.00

Question **1**

Complete

Mark 1.00 out of 1.00

How to create newNode in single Linke list Dynamically?

Select one:

- **a. Node newNode = new Node(int value);**
- **b. Node *newNode = new Node();**
- **c. Node *newNode = new Node(value);**
- **d. None**

Question **2**

Complete

Mark 1.00 out of 1.00

If you are adding a new node at the end of the list, which condition would you use to traverse the list?

Select one:

- **a. while (temp->next == nullptr)**
- **b. while (temp->next != nullptr)**
- **c. while (temp->data != nullptr)**
- **d. while (temp != nullptr)**

Question 3

Complete

Mark 1.00 out of 1.00

What is the Output of the following code

```
void display() {
Node *temp = head;
while (temp!= nullptr) { // Fix: iterate until temp is nullptr
cout << temp->data << " -> ";
temp = temp->next;
}
cout << "NULL" << endl;
int main()
{
SLL slist;
slist.add(10);
slist.add(20);
slist.add(30);
slist.display();
return 0;

}
```

Select one:

- **a. 10 -> 20 -> NULL**
- **b. 10 -> 30 -> NULL**
- **c. 10 -> 20 -> 30 -> NULL**
- **d. 10 -> 20 -> 30**

Question 4

Complete

Mark 1.00 out of 1.00

In which of the following cases is using a linked list more beneficial than an array?

Select one:

- **a. When dynamic memory allocation is required.**
- **b. When the size of the data structure is fixed.**
- **c. When memory space is continuous.**
- **d. When elements need to be accessed frequently by index.**

Question **5**

Complete

Mark 0.00 out of 1.00

Output of the following code

```
void display() {

Node *temp = head;

while (temp->next!= nullptr) {
cout << temp->data << " -> ";

temp = temp->next;

}

cout << "NULL" << endl;

int main()
{

SLL slist;
slist.add(10);

slist.add(20);

slist.add(30);

slist.display();
return 0;

}
```

Select one:

- **a. 10 -> 20 -> 30 -> NULL**
- **b. 10 -> 30 -> NULL**
- **c. 10 -> 20 -> NULL**
- **d. 10 -> 20 -> 30**