Reality Check: Week 10

Word Problem (Question #1)

View the following C++ programming code and generate its output:

```
#include<iostream>
using namespace std;
struct node
{
        int data;
        node *next;
};
class list
{
                private:
                node *head, *tail;
                public:
                list()
                {
                        head=NULL;
                        tail=NULL;
                }
                void createnode(int value)
                {
                        node *temp=new node;
                        temp->data=value;
                        temp->next=NULL;
                        if(head==NULL)
                        {
                                head=temp;
                                tail=temp;
                                temp=NULL;
                        }
                        else
                        {
                                tail->next=temp;
                                tail=temp;
                        }
                }
                void display()
                {
                        node *temp=new node;
                        temp=head;
```

```
        while(temp!=NULL)
        {
                cout<<temp->data<<"\t";
                temp=temp->next;
        }
}
void insert_start(int value)
{
        node *temp=new node;
        temp->data=value;
        temp->next=head;
        head=temp;
}
void insert_position(int pos, int value)
{
        node *pre=new node;
        node *cur=new node;
        node *temp=new node;
        cur=head;
        for(int i=1;i<pos;i++)
        {
                pre=cur;
                cur=cur->next;
        }
        temp->data=value;
        pre->next=temp;
        temp->next=cur;
}
void delete_first()
{
        node *temp=new node;
        temp=head;
        head=head->next;
        delete temp;
}
void delete_last()
{
        node *current=new node;
        node *previous=new node;
        current=head;
        while(current->next!=NULL)
        {
                previous=current;
                current=current->next;
        }
        tail=previous;
```

```
                        previous->next=NULL;
                        delete current;
                }
                void delete_position(int pos)
                {
                        node *current=new node;
                        node *previous=new node;
                        current=head;
                        for(int i=1;i<pos;i++)
                        {
                                previous=current;
                                current=current->next;
                        }
                        previous->next=current->next;
                }
};
int main()
{
        list obj;
        obj.createnode(25);
        obj.createnode(50);
        obj.createnode(90);
        obj.createnode(40);
        cout<<"\n----------------------------------------------\n";
        cout<<"---------------Displaying All nodes---------------";
        cout<<"\n----------------------------------------------\n";
        obj.display();
        cout<<"\n----------------------------------------------\n";
        cout<<"----------------Inserting At End----------------";
        cout<<"\n----------------------------------------------\n";
        obj.createnode(55);
        obj.display();
        cout<<"\n----------------------------------------------\n";
        cout<<"----------------Inserting At Start----------------";
        cout<<"\n----------------------------------------------\n";
        obj.insert_start(50);
        obj.display();
        cout<<"\n----------------------------------------------\n";
        cout<<"-------------Inserting At Particular-------------";
        cout<<"\n----------------------------------------------\n";
        obj.insert_position(5,60);
        obj.display();
        cout<<"\n----------------------------------------------\n";
        cout<<"----------------Deleting At Start----------------";
        cout<<"\n----------------------------------------------\n";
        obj.delete_first();
```

```
        obj.display();
        cout<<"\n----------------------------------------------\n";
        cout<<"-----------------Deleing At End-------------------";
        cout<<"\n----------------------------------------------\n";
        obj.delete_last();
        obj.display();
        cout<<"\n----------------------------------------------\n";
        cout<<"--------------Deleting At Particular--------------";
        cout<<"\n----------------------------------------------\n";
        obj.delete_position(4);
        obj.display();
        cout<<"\n----------------------------------------------\n";
        system("pause");
        return 0;
}
```