UNIVERSITE ABDELMALEK ESSAADI FACULTE DES SCIENCES ET TECHNIQUES DE TANGER
DEPARTEMENT GENIE INFORMATIQUE

# Devoir Final POO C++

**Préparé par :**
**Chaimae Bouassab**
**Licence en :**
**Ingénierie du Développement d'Applications Informatiques**

```cpp
#include <iostream>

class Stack {
private:
    struct Node {
        int data;// data represente l'entier n.
        Node* previous;
    };

    Node* sommet; // Elle pointe vers le dernier élément ajouté à la pile.

public:
 Stack(int x1 = 20) {
        data = x1;
    }
    Stack() : sommet(nullptr) {}

    void operator<<(int data) {
        Node* element = new Node{data, sommet};
        sommet = element;
    }

    int pile_pop(int &data) {
        if (sommet == nullptr) {
            return -1; // La pile va retourner une erreur.
        }

        Node* supp_element = sommet;
        data = supp_element->data;
        sommet = sommet->previous;
        delete supp_element;
```

```
        return 0;
    }

    void affiche() {
        Node* courant = sommet;
        while (courant != nullptr) {
            std::cout << "la valeur : " << courant->data << std::endl;
            courant = courant->previous;
        }
    }
    void pile_clear() {
    int data;
    while (pile_pop(data) == 0) {
        // Continue à dépiler jusqu'à ce que la pile soit vide.
    }
}
};

int main() {
    Stack stack;
    stack.operator<<(12);
    stack.operator<<(1);
    stack.operator<<(2);
    stack.operator<<(3);

    std::cout << "Contenu de la pile : " << std::endl;
    stack.affiche();

    int popped_data;
    int resultat = stack.pile_pop(popped_data);
    if (resultat == 0) {
        std::cout << "Donnée que on va retirer : " << popped_data << std::endl;
    } else {
        std::cout << "La pile est vide." << std::endl;
    }

    return 0;
}
```

```cpp
#include <iostream>

class Stack {
private:
    struct Node {
        int data;
        Node* previous;
    };

    Node* sommet;
 // Elle pointe vers le dernier élément ajouté à la pile.

public:
 Stack(int x1 = 20) {
        data = x1;
    }
    Stack() : sommet(nullptr) {}

    void operator<<(int data) {
        Node* element = new Node{data, sommet};
        sommet = element;
    }

    int pile_pop(int &data) {
        if (sommet == nullptr) {
            return -1;
 // La pile va retourner une erreur.
        }

        Node* supp_element = sommet;
        data = supp_element->data;
        sommet = sommet->previous;
        delete supp_element;

        return 0;
    }

    void affiche() {
        Node* courant = sommet;
        while (courant != nullptr) {
            std::cout << "la valeur : " << courant->data << std::endl;
            courant = courant->previous;
        }
    }
};

int main() {
    Stack stack;
    stack.operator<<(12);
    stack.operator<<(1);
    stack.operator<<(2);
    stack.operator<<(3);

    std::cout << "Contenu de la pile : " << std::endl;
    stack.affiche();

    int popped_data;
    int resultat = stack.pile_pop(popped_data);
    if (resultat == 0) {
        std::cout << "Donnée que on va retirer : " << popped_data << std::endl;
    } else {
        std::cout << "La pile est vide." << std::endl;
    }

    return 0;
}
```