

Correction exam blanc 2

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
class Enseignant {
```

```
private:
```

```
    string nom;
```

```
    string prenom;
```

```
    string email;
```

```
    unsigned int num_bureau;
```

```
    float* notes;
```

```
    size_t nb_notes;
```

```
public:
```

```
    Enseignant() : nom(""), prenom(""), email(""), num_bureau(0), notes(nullptr), nb_notes(0) {}
```

```
    Enseignant(const string& nom, const string& prenom, const string& email, unsigned int num_bureau,  
size_t nb_notes)
```

```
        : nom(nom), prenom(prenom), email(email), num_bureau(num_bureau), nb_notes(nb_notes) {
```

```
        notes = new float[nb_notes]();
```

```
    }
```

```
    Enseignant(const Enseignant& other) : nom(other.nom), prenom(other.prenom), email(other.email),  
num_bureau(other.num_bureau), nb_notes(other.nb_notes) {
```

```
        notes = new float[nb_notes];
```

```
        for (size_t i = 0; i < nb_notes; ++i) {
```

```
        notes[i] = other.notes[i];
    }
}
```

```
~Enseignant() {
    delete[] notes;
}
```

```
void afficher() const {
    cout << "Nom: " << nom << endl
        << "Prénom: " << prenom << endl
        << "Email: " << email << endl
        << "Numéro de bureau: " << num_bureau << endl
        << "Notes: ";
    for (size_t i = 0; i < nb_notes; ++i) {
        cout << notes[i] << " ";
    }
    cout << endl;
}
```

```
void saisir() {
    cout << "Saisir le nom: ";
    cin >> nom;
    cout << "Saisir le prénom: ";
    cin >> prenom;
    cout << "Saisir l'email: ";
    cin >> email;
    cout << "Saisir le numéro de bureau: ";
    cin >> num_bureau;
```

```

    cout << "Combien de notes? ";

    cin >> nb_notes;

    notes = new float[nb_notes];

    for (size_t i = 0; i < nb_notes; ++i) {

        cout << "Note " << i + 1 << ": ";

        cin >> notes[i];

    }

}

```

```

friend bool operator==(const Enseignant& e1, const Enseignant& e2);

};

```

```

bool operator==(const Enseignant& e1, const Enseignant& e2) {

    return e1.nom == e2.nom && e1.prenom == e2.prenom && e1.email == e2.email && e1.num_bureau
    == e2.num_bureau;

}

```

```

class Vacataire : public Enseignant {

private:

    double nombre_hr;

    double salaire_hr;

    static int compteur_id;

    int id;

}

```

```

public:

    Vacataire() : nombre_hr(0), salaire_hr(0), id(++compteur_id) {}

```

```

    Vacataire(const string& nom, const string& prenom, const string& email, unsigned int num_bureau,
    size_t nb_notes,

```

```
        double nombre_hr, double salaire_hr)

: Enseignant(nom, prenom, email, num_bureau, nb_notes), nombre_hr(nombre_hr),
salaire_hr(salaire_hr), id(++compteur_id) {}
```

```
double salaire() const {
    return nombre_hr * salaire_hr;
}
```

```
void afficher() const {
    Enseignant::afficher();
    cout << "Nombre d'heures: " << nombre_hr << endl
        << "Salaire par heure: " << salaire_hr << endl
        << "ID: " << id << endl
        << "Salaire total: " << salaire() << endl;
}
```

```
void saisir() {
    Enseignant::saisir();
    cout << "Saisir le nombre d'heures: ";
    cin >> nombre_hr;
    cout << "Saisir le salaire par heure: ";
    cin >> salaire_hr;
}

};
```

```
int Vacataire::compteur_id = 0;
```

```
int main() {
    Enseignant enseignants[10];
```

```

for (int i = 0; i < 10; ++i) {
    cout << endl << "Saisie des informations de l'enseignant " << i + 1 << ":" << endl;
    enseignants[i].saisir();
}

cout << endl << "Affichage des informations des enseignants:" << endl;
for (int i = 0; i < 10; ++i) {
    enseignants[i].afficher();
}

if (enseignants[0] == enseignants[1]) {
    cout << endl << "Les enseignants 1 et 2 ont les mêmes informations." << endl;
} else {
    cout << endl << "Les enseignants 1 et 2 ont des informations différentes." << endl;
}

Vacataire* vacataire = new Vacataire();
cout << endl << "Saisie des informations du vacataire:" << endl;
vacataire->saisir();
cout << endl << "Affichage des informations du vacataire:" << endl;
vacataire->afficher();
delete vacataire;

return 0;
}

```