Correction exam blanc 1

```
#include <iostream>
#include <string>
#include <vector>

using namespace std;

class Vehicule {
protected:
    string marque;
    string modele;
    int anneeFabrication;
    double kilometrage;

public:
    Vehicule() : marque(""), modele(""), anneeFabrication(0), kilometrage(0.0) {}

    Vehicule(const string& marque, const string& modele, int anneeFabrication, double kilometrage)
        : marque(marque), modele(modele), anneeFabrication(anneeFabrication),
kilometrage(kilometrage) {}

    virtual ~Vehicule() {}

    virtual void saisir() {
        cout << "Saisir la marque: ";
        cin >> marque;
        cout << "Saisir le modèle: ";
        cin >> modele;
        cout << "Saisir l'année de fabrication: ";
```

```
        cin >> anneeFabrication;
        cout << "Saisir le kilométrage: ";
        cin >> kilometrage;
    }

    virtual void afficher() const {
        cout << "Marque: " << marque << endl
             << "Modèle: " << modele << endl
             << "Année de fabrication: " << anneeFabrication << endl
             << "Kilométrage: " << kilometrage << " km" << endl;
    }

    friend ostream& operator<<(ostream& os, const Vehicule& v);
    friend istream& operator>>(istream& is, Vehicule& v);
};

ostream& operator<<(ostream& os, const Vehicule& v) {
    os << "Marque: " << v.marque << endl
       << "Modèle: " << v.modele << endl
       << "Année de fabrication: " << v.anneeFabrication << endl
       << "Kilométrage: " << v.kilometrage << " km";
    return os;
}

istream& operator>>(istream& is, Vehicule& v) {
    cout << "Saisir la marque: ";
    is >> v.marque;
    cout << "Saisir le modèle: ";
    is >> v.modele;
```

```
    cout << "Saisir l'année de fabrication: ";
    is >> v.anneeFabrication;
    cout << "Saisir le kilométrage: ";
    is >> v.kilometrage;
    return is;
}

class Voiture : public Vehicule {
private:
    int nombrePortes;
    string couleur;
    bool estExistante;

public:
    Voiture() : Vehicule(), nombrePortes(0), couleur(""), estExistante(false) {}

    Voiture(const string& marque, const string& modele, int anneeFabrication, double kilometrage, int nombrePortes, const string& couleur, bool estExistante)
        : Vehicule(marque, modele, anneeFabrication, kilometrage), nombrePortes(nombrePortes), couleur(couleur), estExistante(estExistante) {}

    bool estEnCirculation() const {
        return estExistante;
    }

    void saisir() override {
        Vehicule::saisir();
        cout << "Saisir le nombre de portes: ";
        cin >> nombrePortes;
```

```
        cout << "Saisir la couleur: ";
        cin >> couleur;
        cout << "La voiture est-elle existante (1 pour oui, 0 pour non) : ";
        cin >> estExistante;
    }

    void afficher() const override {
        Vehicule::afficher();
        cout << "Nombre de portes: " << nombrePortes << endl
             << "Couleur: " << couleur << endl
             << "En circulation: " << (estExistante ? "Oui" : "Non") << endl;
    }
};

int main() {
    // Création d'objets dynamiques
    Vehicule* v = new Voiture();
    Vehicule* v2 = new Voiture();
    cout << "Saisie des informations pour le premier véhicule :" << endl;
    v->saisir();

    cout << endl << "Saisie des informations pour le deuxième véhicule :" << endl;
    v2->saisir();
    cout << endl << "Affichage des informations pour le premier véhicule :" << endl;
    v->afficher();

    cout << endl << "Affichage des informations pour le deuxième véhicule :" << endl;
    v2->afficher();
```

```
    Voiture* voiture1 = dynamic_cast<Voiture*>(v);
    Voiture* voiture2 = dynamic_cast<Voiture*>(v2);

    if (voiture1) {
        cout << endl << "La première voiture est-elle en circulation ? "
            << (voiture1->estEnCirculation() ? "Oui" : "Non") << endl;
    }

    if (voiture2) {
        cout << "La deuxième voiture est-elle en circulation ? "
            << (voiture2->estEnCirculation() ? "Oui" : "Non") << endl;
    }

    // Libération de la mémoire dynamique
    delete v;
    delete v2;

    return 0;
}
```