ASSIGNMENT 4

METHOD OVERRIDING

```cpp
#include <iostream>
#include <string>

using namespace std;

class Vehicle {
public:
    virtual void start() const {
        cout << "The vehicle is starting." << endl;
    }

    virtual void stop() const {
        cout << "The vehicle is stopping." << endl;
    }
};

class Car : public Vehicle {
public:
    void start() const override {
        cout << "The car's engine is starting." <<endl;
    }
```
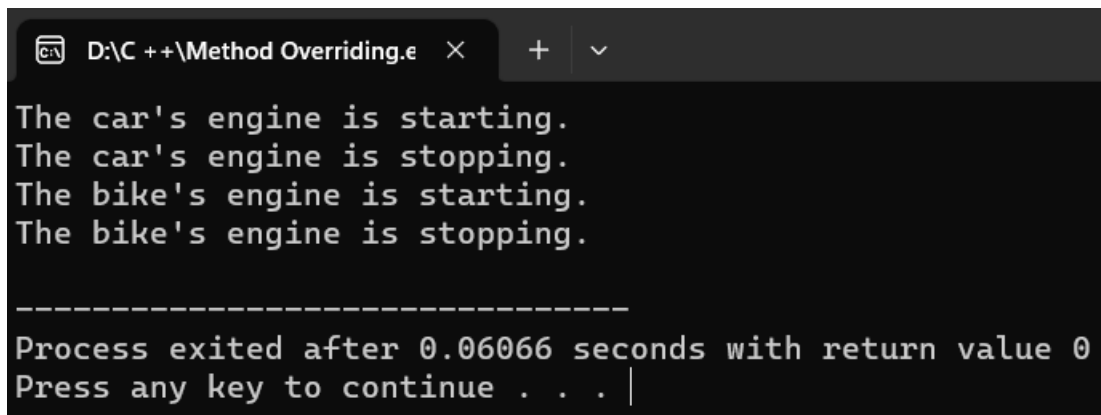
```cpp
        void stop() const override {
            cout << "The car's engine is stopping." <<endl;
        }
};

class Bike : public Vehicle {
public:
    void start() const override {
        cout << "The bike's engine is starting." <<endl;
    }

    void stop() const override {
        cout << "The bike's engine is stopping." <<endl;
    }
};

int main() {
    Vehicle* vehiclePtr;
    Car car;
    Bike bike;

    vehiclePtr = &car;
    vehiclePtr->start();
    vehiclePtr->stop();

    vehiclePtr = &bike;
    vehiclePtr->start();
```

```cpp
    vehiclePtr->stop();

    return 0;
}
```



```
The car's engine is starting.
The car's engine is stopping.
The bike's engine is starting.
The bike's engine is stopping.

--------------------------------
Process exited after 0.06066 seconds with return value 0
Press any key to continue . . .
```

FRIEND FUNCTION

```cpp
#include <iostream>

using namespace std;

class MyNumber {
private:
    int num;

public:
    MyNumber(int n) : num(n) {}

    friend void printNumber(const MyNumber& number);
};
```
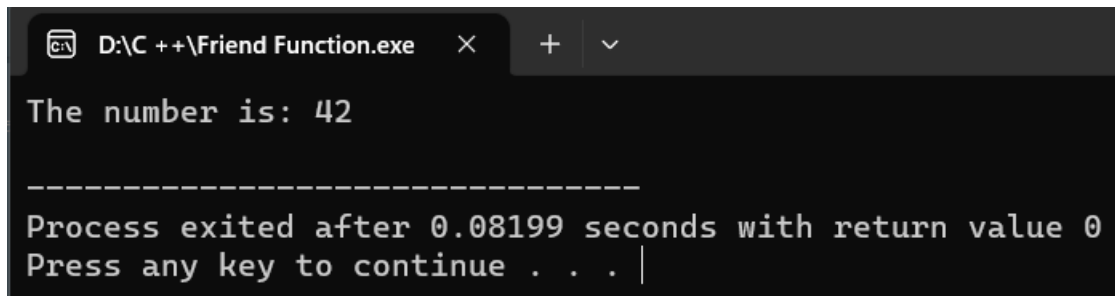
```cpp
void printNumber(const MyNumber& number) {
    cout << "The number is: " << number.num <<endl;
}

int main() {
    MyNumber myNum(42);

    printNumber(myNum);

    return 0;
}
```



```
The number is: 42

--------------------------------
Process exited after 0.08199 seconds with return value 0
Press any key to continue . . .
```