Ex. 6.1b

```
#include <iostream>
using namespace std;
int main()
{
    int *pInt = new int; // claim a piece of memory in Free Store with size of an integer
    *pInt = 9;
    cout << "The value at pInt: " << *pInt << endl;
    delete pInt;
    pInt=0;
    return 0;
}

#include <iostream>
using namespace std;
int main()
{
    int SomeVariable = 5;
    cout << "SomeVariable: " << SomeVariable << "\n";
    int *pVar = & SomeVariable;
    *pVar = 9; //We are not assigning a pointer but the variable pointed by the pointer.
    cout << "SomeVariable: " << *pVar << "\n";
    return 0;
}
```

Exercise 6.1c

```
#include <iostream> //For cout, cin
using namespace std;

class Cat           //Declare the class object
{
    public:
        Cat();      //Constructor
        ~Cat();     //Destructor
        void SetAge (int age);
        int GetAge() const;
        void SetWeight (int weight);
        int GetWeight() const;
    private:
        int itsAge;
        int itsWeight;
};

Cat::Cat()
{
    itsAge = 5;
    itsWeight = 10;
}

Cat::~Cat()
{
    cout << "Before delete, the cat's age is " << itsAge << " and weight is " << itsWeight << ".\n\n";
}

int Cat::GetAge() const
```

```
{
    return itsAge;
}

void Cat::SetAge(int age)
{
    itsAge = age;
}

int Cat::GetWeight() const
{
    return itsWeight;
}

void Cat::SetWeight(int weight)
{
    itsWeight = weight;
}

void catProgram()           //Function for create and modify cat
{
    int input_age = 0;
    int input_weight = 0;
    Cat *Felix = new Cat;   //Create Felix with initial values, Felix is a
pointer of an object of Cat
    cout << "Felix's initial age is " << Felix->GetAge() << " and weight is "
<< Felix->GetWeight() << ".\n\n";

    //Ask user to input age and weight after initialized
    cout << "Please enter Felix age: ";
    cin >> input_age;
    Felix->SetAge(input_age);
    cout << "Please enter Felix weight: ";
    cin >> input_weight;
    Felix->SetWeight(input_weight);
    cout << "Felix's current age is " << Felix->GetAge() << " and weight is "
<< Felix->GetWeight() << ".\n\n";

    char choice = '\0';     //Menu selection
    do
    {
        //Print menu
        cout << "**************************************\n";
        cout << "a) Enter the age and weight again\n";
        cout << "b) Destroy Felix and create again\n";
        cout << "c) Quit the program\n";
        cout << "**************************************\n";
        cout << "What do you want to do? (a, b or c) ";
        cin >> choice;

        switch (choice)
        {
            case 'a':
            case 'A':
                //Ask user to input age and weight again
                cout << "Please enter Felix age again: ";
                cin >> input_age;
                Felix->SetAge(input_age);
                cout << "Please enter Felix weight again: ";
                cin >> input_weight;
                Felix->SetWeight(input_weight);
                cout << "Felix's age is " << Felix->GetAge() << " and weight
is " << Felix->GetWeight() << ".\n\n";
                break;

            case 'b':
            case 'B':
                delete Felix;       //Destroy Felix object
                Felix = new Cat;    //Create Felix again
                break;
            case 'c':
            case 'C':
                delete Felix;       //Destroy Felix object
                cout << "Goodbye!!\n\n";
                break;
            default:
                cout << "Invalid input!!!\n\n";
        }
    }while (!(choice == 'c' || choice == 'C'));
}

void main()
{
    char choice = '\0';         //Menu selection
    bool run_flag = true;       //Quit program when it is false

    do
    {
        cout << "Do you want to create Felix? (y/n) ";
        cin >> choice;

        switch (choice)
        {
            case 'y':
            case 'Y':
                catProgram();   //Call function for cat
                run_flag = false;
                break;
            case 'n':
            case 'N':
                cout << "Bye!!\n\n";
                run_flag = false;
                break;
            default:
                cout << "Invalid input!!!\n\n";
                break;
        }
    }while (run_flag);
}
```