Object Oriented Programming Lab Evaluation 4

1. The `Car` Class

- Create a simple class `Car` with **three** attributes: **company**, **model**, **highestSpeed**.
- Write a C++ program to initiate 3 objects of the `Car` class and set these attributes to your favorite car. Then, print out the company, model, and highest speed of the `Car` object.

Ans.: Here is a C++ program that satisfies the given conditions:

```
#include <iostream>
using namespace std;

class Car
{
    string company;
    string model;
    int highestSpeed;

public:
    Car(string com, string m, int hs)
    {
        company = com;
        model = m;
        highestSpeed = hs;
    }

    void displayInfo(int i)
    {
        cout << endl;
        cout << "Car " << i << endl;
        cout << "Model    : " << model << endl;
        cout << "Company  : " << company << endl;
        cout << "Max Speed: " << highestSpeed << "kmph" <<
endl;
    }
};

int main()
{
    Car c1("Honda", "Civic", 137);
    Car c2("KIA", "Carnival MPV", 190);
    Car c3("Toyota", "Avanza", 180);

    c1.displayInfo(1);
    c2.displayInfo(2);
    c3.displayInfo(3);
    return 0;
}
```

Output: The above C++ code yields the following output in the terminal:

```
Car 1
Model    : Civic
Company  : Honda
Max Speed: 137kmph

Car 2
Model    : Carnival MPV
Company  : KIA
Max Speed: 190kmph

Car 3
Model    : Avanza
Company  : Toyota
Max Speed: 180kmph
```

2. The `BankAccount` Class

- Define a class `BankAccount` with **private attributes**: `accountNumber` and `balance`. Include a **constructor** that initializes both attributes. Implement getter methods for `accountNumber` and `balance`, and a setter method for `balance`.

- Write a code snippet to create an instance of `BankAccount` with an `accountNumber` of your Student ID and an initial balance of **1000/=**. Use the setter method to deposit **500/=** into the account. Then, retrieve and print the balance and accountNumber using the getter methods.

Ans.: Here is a C++ program that satisfies the given conditions:

```cpp
#include <iostream>
using namespace std;

class BankAccount
{
    string accountNumber;
    float balance;

public:
    BankAccount(string accno, float b)
    {
        accountNumber = accno;
        balance = b;
    }

    string getAccNumber()
    {
        return accountNumber;
    }

    float getBalance()
    {
        return balance;
    }

    void deposit(float amount)
    {
        balance += amount;
    }
};

int main()
{
    BankAccount account("20245103408", 1000);
    account.deposit(500);

    cout << "Account Number : " << account.getAccNumber()
<< endl;
    cout << "Current Balance: " << account.getBalance() <<
"/=" << endl;
    return 0;
}
```

Output: The above C++ code yields the following output in the terminal:

```
Account Number : 20245103408
Current Balance: 1500/=
```

3. The `Game` Class

- Create a simple class `Game` with **two attributes**: `kill_count`, `death_count`.
- First, make **two objects** of this class using a **constructor**. Then, make a **non-member function** where the function takes another object, sets the `kill_count` and `death_count` attributes' value, and returns the object to the main function.
- Lastly, print all the object values.

Ans.: Here is a C++ program that satisfies the given conditions:

```cpp
#include <iostream>
using namespace std;

class Game
{
    int kill_count;
    int death_count;

public:
    Game(int k, int d) : kill_count(k), death_count(d)
    {
    }

    void setInfo(int k, int d)
    {
        kill_count = k;
        death_count = d;
    }

    void getInfo(int n)
    {
        cout << "Game " << n << endl;
        cout << "Kill Count : " << kill_count << endl;
        cout << "Death Count: " << death_count << endl;
        cout << endl;
    }
};

Game reset(Game g, int k, int d)
{
    g.setInfo(k, d);
    return g;
}

int main()
{
    Game g1(0, 0);
    g1 = reset(g1, 24, 48);

    Game g2(30, 20);
    g2 = reset(g2, 10, 5);

    g1.getInfo(1);
    g2.getInfo(2);
    return 0;
}
```

Output: The above C++ code yields the following output in the terminal:

```
Game 1
Kill Count : 24
Death Count: 48

Game 2
Kill Count : 10
Death Count: 5
```

4. The `Branch` Class

Suppose you are a bank manager.

- Define a class `Branch` with **three attributes**: `branch_name`, `branch_location`, and `employee[10]` (array of objects). Use the get and set methods to set the values of the objects and print them.

- Take a user input of `branch_location` (**string**). Now match which object matches with the new branch name and print all the attributes related to that object.

Ans.: Here is a C++ program that satisfies the given conditions:

```
#include <iostream>
#define employees 10
#define branches 2
using namespace std;

class Person
{
    string name;
    string position;

public:
    Person()
    {
    }

    Person(string n, string p)
    {
        name = n;
        position = p;
    }

    string getName()
    {
        return name;
    }

    string getPosition()
    {
        return position;
    }
};

class Branch
{
    string branch_name;
    string branch_location;
    Person employee[employees];

public:
    Branch(string bn, string bl, Person e[employees])
    {
        branch_name = bn;
        branch_location = bl;

        for (int i = 0; i < employees; i++)
            employee[i] = e[i];
    }

    string getLocation()
    {
        return branch_location;
    }

    void getInfo()
    {
        cout << "Branch Name     : " << branch_name << endl;
        cout << "Branch Location: " << branch_location << 
endl;
        cout << "Employees:" << endl;
        for (int i = 0; i < employees; i++)
            cout << " - " << employee[i].getName() << " "
```

```
<< employee[i].getPosition() << endl;
    }
};

int main()
{
    Person team1[employees] = {
        Person("Shayan", "TR"),
        Person("Munna", "E"),
        Person("Simon", "E"),
        Person("Sharmin", "E"),
        Person("Fatema", "E"),
        Person("Ahona", "E"),
        Person("Rubaiyat", "E"),
        Person("Rebeka", "E"),
        Person("Bulbul", "E"),
        Person("Jamal", "HR")};

    Person team2[employees] = {
        Person("Shahriar", "TR"),
        Person("Rabyea", "E"),
        Person("Rumon", "E"),
        Person("Ronny", "E"),
        Person("Minhaj", "E"),
        Person("Azmain", "E"),
        Person("Himu", "E"),
        Person("Minu", "E"),
        Person("Mobarak", "E"),
        Person("Adnan", "HR")};

    Branch branch[branches] = {
        Branch("BRAC Bank", "Mirpur 11", team1),
        Branch("IDFC Bank", "Motijheel", team2)};

    string searchLocation;
    cout << "Enter Branch Location: " << endl;
    cin >> searchLocation;
    cout << endl;
    for (int i = 0; i < branches; i++)
        if (branch[i].getLocation() == searchLocation)
            branch[i].getInfo();

    return 0;
}
```

Input:

```
+ Enter Branch Location:
  Motijheel
```

Output: The above C++ code yields the following output in the terminal:

```
Branch Name    : IDFC Bank
Branch Location: Motijheel
Employees:
 - Shahriar TR
 - Rabyea E
 - Rumon E
 - Ronny E
 - Minhaj E
 - Azmain E
 - Himu E
 - Minu E
 - Mobarak E
 - Adnan HR
```

Code

You can find all the code snippets **here**.