1. Identify three objects that might belong to each of the following classes?

a. automobile

b. NovelAuthor

c. CollegeCourse

SOLUTION:

a. Mclaren F1
b. Jeffrey Archer
c. Linear Algebra

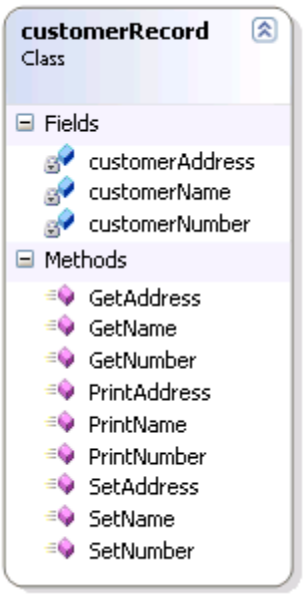
d. Design a class named customerRecord that holds a customer number, name and address include methods to see the values for each data field and print the values for each data field create the class diagram and write the pseudocode that defines the class.

SOLUTION:

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

class customerRecord
{
	private:
		string customerName;	// Holds customer name
		string customerNumber;	// Holds alphanumeric customer number
		string customerAddress;	//  Holds customer Address

	public:
		// Method to get customer name
		string GetName()
		{
			return customerName;
		}
		// Method to get customer number
		string GetNumber()
		{
			return customerNumber;
		}
		// Method to get customer Address
		string GetAddress()
		{
			return customerAddress;
		}

		void SetName(string Name)
		{
			customerName = Name;
		}

		void SetNumber(string Number)
```

```
	{
		customerNumber = Number;
	}

	void SetAddress(string Address)
	{
		customerAddress = Address;
	}

	//Method to print customer name
	void PrintName()
	{
		cout << "Customer Name: " << customerName << endl;
	}
	// Method to print customer number

	void PrintNumber()
	{
		cout << "Customer Number: " << customerNumber << endl;
	}

	// Method to print customer Address

	void PrintAddress()
	{
		cout << "Customer Address: " << customerAddress << endl;
	}

};
```

CLASS DIAGRAM:



PSEUDOCODE:

```
Declare customerRecord
Initialize Customer Name
Initialize Customer Number
Initialize Customer Address
Print Customer Name
Print Customer Number
Print Customer Address
```

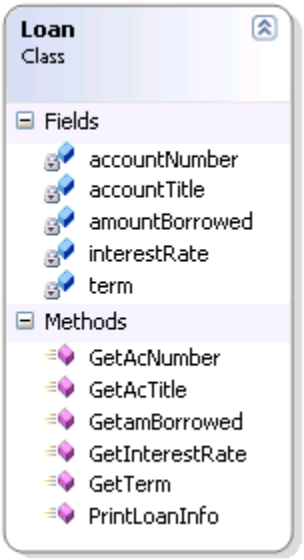
3. Design a class named loan, that holds an account number, name of account holder, amount borrowed, term and interest rate. include methods to see values for each data field and a method that prints all the loan information. create the class diagram and write the pseudo-code that defines the class.

SOLUTION:

```
class Loan
{
private:
      string accountNumber;  // Account Number
      string accountTitle;   // Name of the account holder
      double amountBorrowed;  // Amount borrowed
      int term;               // Term of loan
      float interestRate;     // interest rate

public:
    // Method to get Account Number
      string GetAcNumber()
      {
            return accountNumber;
      }
      // Method to get Account Title (Name of Account Holder
      string GetAcTitle()
      {
            return accountTitle;
      }
      double GetamBorrowed()
      {
            return amountBorrowed;
      }
      int GetTerm()
      {
            return  term;
      }
      float GetInterestRate()
      {
            return interestRate;
      }

      void PrintLoanInfo()
      {
            cout << "Account Title: " << accountTitle << endl;
            cout << "Account Number: " << accountNumber<< endl;
            cout << "Amount Borrowed: " << amountBorrowed << endl;
            cout << "Loan Term: " << term << endl;
```

```
            cout << "Interest Rate: " << interestRate << endl;
      }
};
```

CLASS DIAGRAM:



PSEUDOCODE:

```
Declare Loan
Initialize Account Holder Name
Initialze Account Number to
Initialze Amount Borrowed
Initialzie interest Rate
Initialize term

Print User Loan Information
```

4. Design a class name circle with fields named radius, area and diameter include a constructor that sets the radius to 1. include get methods for each field but include a set method only for the radius when the radius is set calculate the diameter (twice the radius) and the area(the radius squared time pi, which is approximately 3.14) create the class diagram and write the pseudocode that defines the class. B) design an application program that declares two circles set the radius of one manually but allow the other to use the default value supplied by the constructor then display each circles values.

SOLUTION:

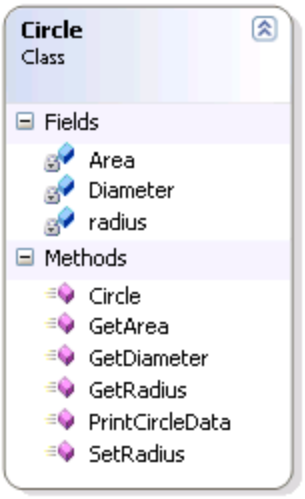
```
class Circle
{
private:
      double radius;
      double Area;
      double Diameter;
public:
```

```
	Circle()
	{
		radius = 1;
		Diameter = 2 * radius;
		Area = 3.1415926 * radius * radius;
	}
	// Get Radius
	double GetRadius()
	{
		return radius;
	}
	// Get Area
	double GetArea()
	{
		return Area;
	}
	// Get Diameter
	double GetDiameter()
	{
		return Diameter;
	}
	// Set Radius
	void SetRadius(double Rad)
	{
		radius = Rad;
		Diameter = 2 * radius;
		Area = 3.1415926 * radius * radius;
	}

	void PrintCircleData()
	{
		cout << "Radius: " << radius << endl;
		cout << "Diameter: " << Diameter << endl;
		cout << "Area: " << Area << endl;
	}

};

int main(void)
{
	Circle c1,c2;
	c1.PrintCircleData();
	c2.SetRadius(2.5);
	c2.PrintCircleData();

	getche();

	return 0;
}
```

Output:

```
Radius: 1
Diameter: 2
Area: 3.14159
Radius: 2.5
Diameter: 5
Area: 19.635
```

CLASS DIAGRAM:



PSUEDOCODE:

Delare First Circle
Declare Second Circle
Print First Circle's Radius
Print First Circle's Diameter
Print First Circle's Area
Initialize Second Circle's Radius to 2.5
Print Second Circle's Radius
Print Second Circle's Diameter
Print Second Circle's Area

5. Design a class named girlscout with fields that hold a name troop number and dues owed include get and set methods for each field include a static method that displays the girl scout motto (to obey the girl scout law) include three overloaded constructors as follows: a default constructor that sets the name to xxx and the numeric fields to 0 a constructor that allows you to pass values for all three fields a constructor that allows you to pass a name and troop number but sets dues owed to 0 Create the class diagram and write the pseudo-code that defines the class B) design an application program that declares three girlscout objects using a different constructor version with each object display each girlscout values then display motto.

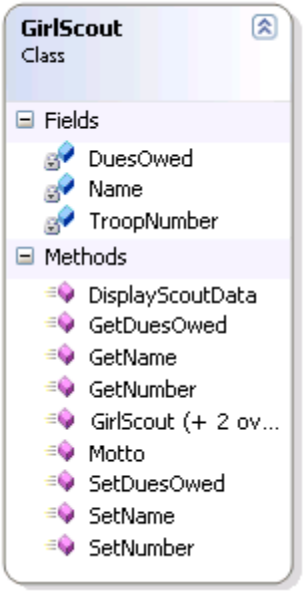
```
class GirlScout
{
private:
	string Name;
	string TroopNumber;
	double DuesOwed;

public:

	// Contructor 1
	GirlScout()
	{
		Name = "xxx";
		TroopNumber = "xxx";
		DuesOwed = 0;
	}
	// Constructor 2
	GirlScout(string name, string tnumber, double dues)
	{
		Name = name;
		TroopNumber = tnumber;
		DuesOwed = dues;
	}
	// Constructor 3
	GirlScout(string name, string tnumber)
	{
		Name = name;
		TroopNumber = tnumber;
		DuesOwed = 0;
	}

	string GetName()
	{
		return Name;
	}

	string GetNumber()
	{
		return TroopNumber;
	}

	double GetDuesOwed()
	{
		return DuesOwed;
```

```
	}

	void SetName(string name)
	{
		Name = name;
	}

	void SetNumber(string number)
	{
		TroopNumber = number;
	}

	void SetDuesOwed(double dues)
	{
		DuesOwed = dues;
	}

	static void Motto()
	{
		cout << "Motto: To obey the girl scout law" << endl;
	}

	void DisplayScoutData()
	{
		cout << "Scout Name: " << Name << endl;
		cout << "Troop Number: " << TroopNumber << endl;
		cout << "Dues: " << DuesOwed << endl;
	}
};

int main(void)
{
	GirlScout gs1;
	gs1.DisplayScoutData();
	GirlScout::Motto();
	GirlScout gs2("Maria","12754");
	gs2.DisplayScoutData();
	GirlScout::Motto();
	GirlScout gs3("Summers","36434",312);
	gs3.DisplayScoutData();
	GirlScout::Motto();

	getche();

	return 0;
}
```

Output:

```
Scout Name: xxx
Troop Number: xxx
Dues: 0
Motto: To obey the girl scout law
Scout Name: Maria
Troop Number: 12754
Dues: 0
Motto: To obey the girl scout law
Scout Name: Summers
Troop Number: 36434
Dues: 312
Motto: To obey the girl scout law
```

CLASS DIAGRAM:



PSEUDOCODE:

Declare first GirlScout

Print First GirlScout's Name

Print First GirlScout's Troop Number

Print First GirlScout's Dues

Initialize second GirlScouts Name and Troop Number

Print Second GirlScout's Name

Print Second GirlScout's Troop Number

Print Second GirlScout's Dues
Declare third GirlScout
Initialize third GirlScout's Name, Troop Number, Dues Borrowed
Print third GirlScout's Name
Print third GirlScout's Troop Number
Print third GirlScout's Dues