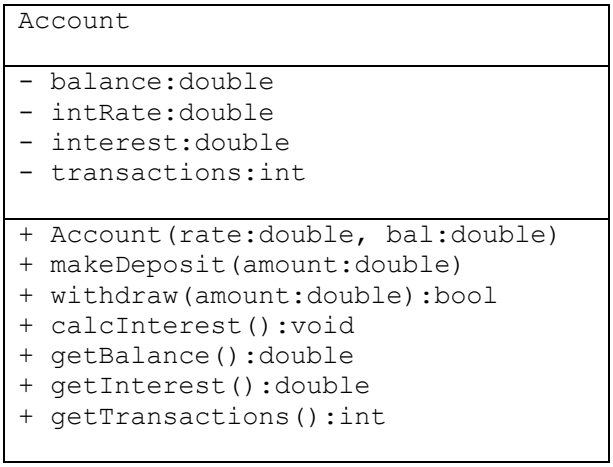
Lab 04
Object-oriented Design and Programming

Section 1: Guess program outputs.

1.

Given the following UML Class Diagram design for the Account class and the program that uses it. Guess program outputs.




```
#include <iostream>
#include <string>
#include <iomanip>

using namespace std;

class Account
{
 private:
   double balance;
   double intRate;
   double interest;
   int transactions;

 public:
      Account(double rate = 0.045, double bal = 0.0)
   {  balance = bal;  intRate = rate;
      interest = 0.0; transactions = 0;
   }

   void makeDeposit(double amount)
   {  balance += amount;
      transactions++;
   }

   bool withdraw(double amount);

   void calcInterest()
   {  interest = balance * intRate;
      balance += interest;
   }
```

```
   double getBalance()
   {  return balance;
   }

   double getInterest()
   {  return interest;
   }

   int getTransactions()
   {  return transactions;
   }
};

bool Account::withdraw(double amount)
{
      if (balance < amount)
            return false;   // Not enough in the account
      else
      {
            balance -= amount;
            transactions++;
            return true;
      }
}

void displayMenu();
char getChoice(char max);
void makeDeposit(Account &account);
void withdraw(Account &account);

int main()
{
      const char MAX_CHOICE = '7';
      Account savings;
      char choice;

      cout << fixed << showpoint << setprecision(2);
      do
      {
            displayMenu();
            choice = getChoice(MAX_CHOICE);
            switch(choice)
            {
                  case '1': cout << "The current balance is $";
                              cout << savings.getBalance() << endl;
                                break;
                  case '2': cout << "There have been ";
                              cout << savings.getTransactions()
                                    << " transactions.\n";
                                break;
                case '3': cout << "Interest earned for this period: $";
                              cout << savings.getInterest() << endl;
                                break;
                case '4': makeDeposit(savings);
                              break;
                case '5': withdraw(savings);
                              break;
                case '6': savings.calcInterest();
                              cout << "Interest added.\n";
          }
      } while(choice != '7');
```

```
    return 0;
}

void displayMenu()
{
   cout << "\n\n                MENU\n\n";
   cout << "1) Display the account balance\n";
   cout << "2) Display the number of transactions\n";
   cout << "3) Display interest earned for this period\n";
   cout << "4) Make a deposit\n";
   cout << "5) Make a withdrawal\n";
   cout << "6) Add interest for this period\n";
   cout << "7) Exit the program\n\n";
   cout << "Enter your choice: ";
}

char getChoice(char max)
{
      char choice = cin.get();
      cin.ignore();

      while (choice < '1' || choice > max)
      {
            cout << "Choice must be between 1 and " << max << ". "
                 << "Please re-enter choice: ";
            choice = cin.get();
            cin.ignore();
      }
      return choice;
}

void makeDeposit(Account &account)
{
   double dollars;

   cout << "Enter the amount of the deposit: ";
   cin  >> dollars;
   cin.ignore();
   account.makeDeposit(dollars);
}

void withdraw(Account &account)
{
   double dollars;

   cout << "Enter the amount of the withdrawal: ";
   cin  >> dollars;
   cin.ignore();
   if (!account.withdraw(dollars))
         cout << "ERROR: Withdrawal amount too large.\n\n";
}
```

Section 2: Review Questions and Exercises

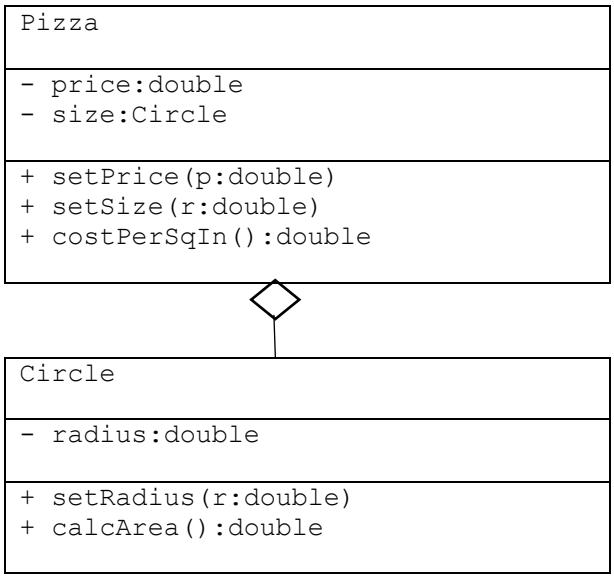
1. What is a problem domain?

2. How do you identify the potential classes in a problem domain description?

Look at the following description of a problem domain:
A doctor sees patients in her practice. When a patient comes to the practice, the doctor performs one or more procedures on the patient. Each procedure performed has a description and a standard fee. As patients leave, they receive a statement that shows their name and address, as well as the procedures that were performed and the total charge for the procedures. Assume that you are creating an application to generate a statement that can be printed and given to the patient.
3. Identify all of the potential classes in this problem domain.
4. Refine the list to include only the necessary class or classes for this problem.
5. Identify the responsibilities of the class or classes that you identified in step 4.

Section 3: Programming Challenges

1. Write a complete program based on the following the UML class diagram, the main() function and the program output.

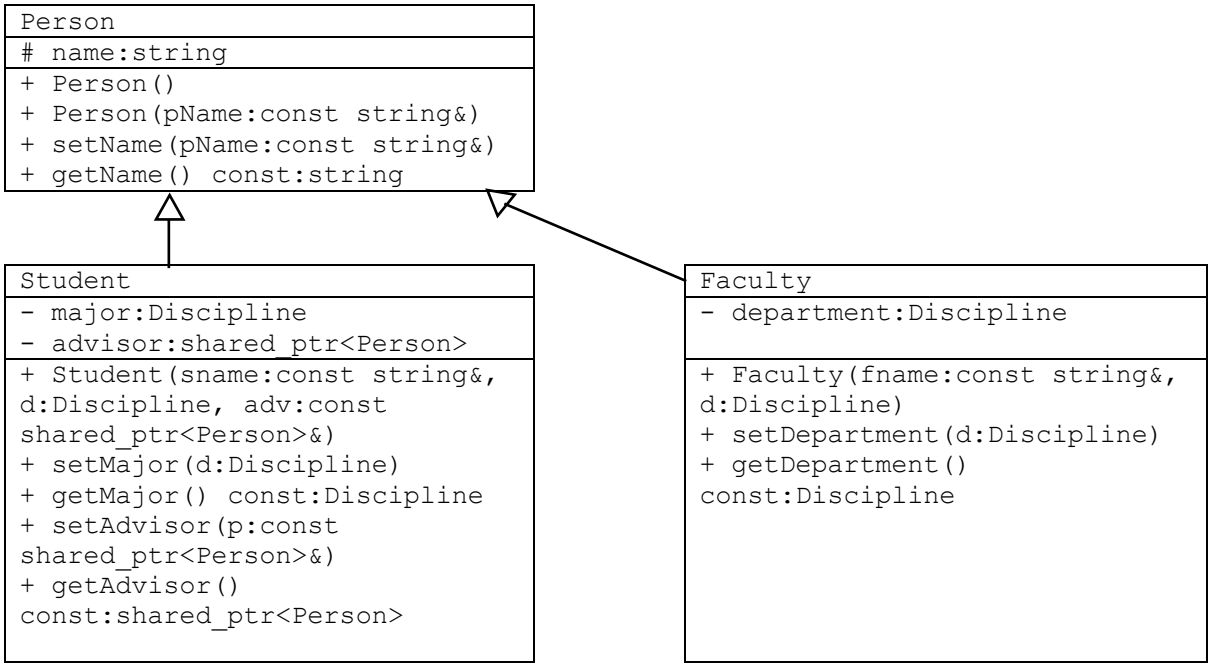



```
int main()
{
  Pizza myPizza;
  myPizza.setPrice(314.159);
  myPizza.setSize(10);
  cout << "Price per square inch $"<< myPizza.costPerSqIn();

  return 0;
}
```

Program Output
Price per square inch $1

2. Write a complete program based on the following the UML class diagram, the main() function and the program output.




```
int main()
{
    shared_ptr<Faculty>
    prof =  make_shared<Faculty>("Indiana Jones",
                                 Discipline::ARCHEOLOGY);
    shared_ptr<Student>
    st =  make_shared<Student>("Sean Bolster",
                               Discipline::ARCHEOLOGY, prof);

    cout << "Professor " << prof->getName() << " teaches "
         << dName[static_cast<int>(prof->getDepartment())]
         << "." << endl;

    shared_ptr<Person> pAdvisor = st->getAdvisor();
    cout << st->getName() << "\'s advisor is "
         << pAdvisor->getName() << ".";
    cout << endl;

    return 0;
}
```

Program Output

Professor Indiana Jones teaches Archeology.
Sean Bolster's advisor is Indiana Jones.