Lab # 3: Artificial Intelligence (CSC-1071)

Basic Information			
Registration#		**Name**	
Total Marks	10	**Marks Obtained**	
Tools:	Any Interpreter of your choice		
Objectives	1. Data Encapsulation 2. Operator Overloading	3. GUI	

Classes & Objects

1. Implement Loan Class with following specs and use this class:
Private Properties with default values: annualInterestRate = 2.5, numberOfYears = 1, loanAmount = 1000, borrower = " " Functions: getAnnualInterestRate(self), getNumberOfYears(self), getLoanAmount(), getBorrower(self), setAnnualInterestRate(self, annualInterestRate), setNumberOfYears(self, numberOfYears), setLoanAmount(self, loanAmount), setBorrower(self, borrower), getMonthlyPayment(self): monthlyInterestRate = self.__annualInterestRate / 1200, monthlyPayment = self.__loanAmount * monthlyInterestRate / (1 - (1 (1 + monthlyInterestRate) ** (self.__numberOfYears * 12))) return monthlyPayment
getTotalPayment(self):totalPayment = self.getMonthlyPayment() * self.__numberOfYears * 12 return totalPayment

2. Implement BMI Class with following :
Private Data Initializer: def __init__(self, name, age, weight, height):
Functions: getBMI(self):, getStatus: bmi = self.getBMI(), if bmi < 18.5: return "Underweight" etc., getName(self) getAge(self), getWeight(self), self.__weight, getHeight(self),

Operator overloading & Inheritence

3. Overload -, * and / for complex class

4. *(*RationalNumber *Class)* Create a class RationalNumber (fractions) with the following capabilities:
a) Create a constructor that prevents a 0 denominator in a fraction, reduces or simplifies fractions that are not in reduced form and avoids negative denominators.
b) Overload the addition, subtraction, multiplication and division operators for this class.
c) Overload the relational and equality operators for this class.

5. *(*Polynomial *Class)* Develop class Polynomial. The internal representation of a Polynomial is an array of terms. Each term contains a coefficient and an exponent, e.g., the term2*x*4 has the coefficient 2 and the exponent 4. Develop a complete class containing proper constructor and destructor functions as well as *set* and *get* functions. The class should also provide the following overloaded operator capabilities:
a) Overload the addition operator (+) to add two Polynomials.
b) Overload the subtraction operator (-) to subtract two Polynomials.
c) Overload the assignment operator to assign one Polynomial to another.
d) Overload the multiplication operator (*) to multiply two Polynomials.
e) Overload the addition assignment operator (+=), subtraction assignment operator (-=), and multiplication assignment operator (*=).

GUI Applications

6. Implement Question 1 using appropriate GUI

7. Implement Question 2 using appropriate GUI

8. Implement 8 Puzzle grid with a Shuffle Button