Name ________________________

CIS 180 – Object-Oriented Programming I
Practice Examination #1

1) (5 pts) Give the binary representation of the decimal number 11: _____________
2) (5 pts) Give the binary representation of the decimal number 22: _____________
3) (5 pts) Give the sum of the binary numbers 0101 and 0011 in binary: __________
4) (5 pts) What keyword in Java is used to define inheritance? __________________
5) (5 pts) What operator or keyword is used in Java to create objects? ____________
6) (5 pts) Give a brief definition of the term *inheritance*:
7) (5 pts) Give a brief explanation of what is meant by *type polymorphism*:
8) (5 pts) Briefly describe the relationship between *classes* and *objects* in an object oriented program:
9) (5 pts) Briefly describe the relationship between *methods* and *messages* in an object oriented program:

10) Consider a java *Account* class for bank account objects:

- Each account object should have an account number and a balance.
- There should be a constructor with account number and balance parameters.
- There should be accessor methods *getAccountNumber*, *getBalance,* and *setBalance*.
- There should be a *deposit* method with an amount parameter that adjusts the balance according to the amount deposited.
- There should be a *withdraw* method for withdrawing money from the account.

The constructor method and the deposit and with withdraw methods should print "receipts" on the console. If a request is made to withdraw more money than is in the account, an error message should be printed and the account balance left unchanged.

The following example shows how the Account class could be used in an application:

```java
public class BankingApplication
{
	public static void main(String[] args)
	{
		Account a1 = new Account(101, 500);
		a1.deposit(50);
		a1.withdraw(100);
		a1.withdraw(800);
		a1.deposit(20);
	}
}
```

The example application shown above should produce the following output:

```
Opened account 101 with initial balance of $500.00

Deposited $50.00 in account 101
The new balance is: $550.00

Withdrew $100.00 from account 101
The new balance is: $450.00

Error: Insufficient funds for
withdrawal of $800.00 from account 101

Deposited $20.00 in account 101
The new balance is: $470.00
```

a. (10 pts) Draw a UML class diagram for the Account class showing its attributes and methods. Include all details (types, parameters, accessibility):

b. (25 pts) Write a Java implementation of the Account class:

9) Consider an *OverdraftAccount* class. Overdraft accounts are similar to regular accounts except that they can be overdrawn (have a negative balance) up to a preset limit. The constructor for OverdraftAccounts should take the account number, initial balance, and overdraft limit as parameters.

   a. (5 pts) Add the OverdraftAccount class to the UML class diagram you drew in problem 10. Show the relationship between the two classes on the diagram.

   b. (15 pts) Write a Java implementation of the OverdraftAccount class: