Assignment 5
(**Due date: 11/19/2008/Wednesday, in class**)

Your name:	Date:

Provide brief answers to the following questions (you may check the textbook Section Questions & Exercises for solutions to some of the questions in the **Chapters 5 (Algorithms) & 6 (Programming Languages)**, read the textbook and lecture slides for other questions):

1. Give a brief definition of algorithm. Summarize the distinctions between a process, an algorithm, and a program.

2. In what sense do the steps described by the following list of instructions fail to constitute an algorithm?
   1) Step 1. Take a coin out of your pocket and put it on the table.
   2) Step 2. Return to Step 1.

3. In what sense is the construction of procedures the construction of primitives?

4. The Euclidean algorithm finds the greatest common divisor of two positive integers X and Y by the following process:

   As long as the value of neither X nor Y is zero, continue dividing the larger of the values by the smaller and assigning X and Y the values of the divisor and remainder, respectively. (The final value of X is the greatest common divisor.)

Express this algorithm in pseudocode like that in the textbook.

5. Convert the pseudo code routine

```
Z ← 0;
X←1;
While (x < 6) do
   (Z←Z+X;
    X←X+1)
```

to an equivalent routine using a repeat statement.

6. What is the difference between a formal programming language and a pseudocode?

7. Identify the termination condition in the following iterative statements:
   1) while (Count < 5) and (Total < 56) do ( )
   2) repeat ( ) until (Count = 1)

8. Identify the body of the following loop structure and count the number of times it will be executed.

```
Count←1;
while (Count not 7) do
   (print the value assigned to Count and
    Count ←Count + 3)
```

9. What is the difference between an assembler and a compiler?

10. Why is the use of a constant considered better programming style than the use of a literal?

11. List some common data types and describe how to use them for representing personal information such as ages, height, name…?

12. Identify some common control structures found in imperative and object-oriented programming languages.

13. What is the difference between a global variable and a local variable?

14. What is the difference between a procedure and a function?

15. When writing in modern programming languages, programmers tend to use verbs for names of procedures and nouns for names of functions. Why?

16. Draw the parse tree for the expression: x * y + x + z. (* is multiplication operator).

17. What is the difference between an object and a class?

18. Suppose t he classes PartTimeEmployee and FullTimeEmployee inherited the properties of the class Employee. What are some features that you might expect to find in each class?

19. Which of the statements R, S, T, U and V are logical consequences of the collection of statements (~R OR T OR S), (~S OR V), (~V OR R), (U OR ~S), (T OR U), and (S OR V)? (~ means NOT operation).

20. Figure 6.2 (available from slides for the Chapter 6) provides a diagram that describes the evolution of programming paradigms. Put each of the programming languages in that diagram in the following table:

	Functional	Object-oriented	Imperative	Declarative
1950				
1960				
1970				
1980				
1990				
2000				

21. Translate the following *for* statement into an equivalent program segment using the while statement in our pseudocode: **for (int x = 2; x < 8; ++x) {…..}**

22. Summarize the distinction between a machine language and an assembly language.

23. Summarize the distinction between declarative statements and imperative statements.

==========**Important Notes**==========

- Homework can be hand-written or typewritten.
- Put all your solutions in the same order as the above questions.
- Use this question paper as **cover page and staple them together**.