P. 24 Q. 2) What are the two main components of a computer?

Hardware and software

4) Why is secondary storage needed?

Because memory (RAM) is limited and volatile. Anything stored in RAM is lost on a reboot, and RAM is more expensive per byte than other forms of storage.

6) What are the two types of programs?

System programs and application programs

8) What is an object program?

The machine language version of the high-level language program

10) What is linking?

Combining an object program with other programs in the library using a linker

12) What is an algorithm?

A step-by-step problem solving process in which a solution is arrived at in a finite amount of time.

14) What are the advantages of problem analysis and algorithm design over directly writing a program in a high-level language?

Knowing a problem thoroughly and then outlining an algorithm allows a programmer to have an idea how to structure his code before writing a single line.

Chapter Project assignments 20%

Chapter Quizzes 20%

Attendance 10%

Midterm Exam 20%

Final Exam 30%

Input Process Output

Chapter Project assignments % \* 20 final grade

Chapter Quizzes % \* 20

Attendance % \* 10

Midterm Exam % \* 20

Final Exam % \* 30

Final grade = sum of above

| **Name** | **Date Created** | **Author(s)** | **Description** |
| --- | --- | --- | --- |
| Machine Code |  |  |  |
| FORTRAN | 1956 | John W. Backus | a general-purpose, imperative programming language that is especially suited to numeric computation and scientific computing |
| COBOL | 1959 | Howard Bromberg, Howard Discount, Vernon Reeves,  Jean E. Sammet, William Selden, and Gertrude Tierney | a compiled English-like computer programming language designed for business use |
| C | 1972 | Dennis Ritchie | a general-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion, while a static type system prevents many unintended operations |
| JAVA | 1995 | James Gosling | general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible |
| Python | 1991 | Guido van Rossum | A high-level, general-purpose, interpreted, dynamic programming language |
| Perl | 1987 | Larry Wall | a family of high-level, general-purpose, interpreted, dynamic programming languages |
| Ruby | 1995 | Yukihiro Matsumoto | a dynamic, reflective, object-oriented, general-purpose programming language |
| Rexx | 1979 | Mike Cowlishaw | a structured, high-level programming language designed for ease of learning and reading |
| PASCAL | 1968 | Niklaus Wirth | Was created with the intention to make an efficient language based on structured programming |