Ans1) Relocatable code is software whose execution address can be changed. A relocatable program might run at address 0 in one instance, and at 10000 in another."Relocatable" means the assembler builds code chunks and specifies RAM needs that can be placed wherever the linker finds room for them. Relocatable machine code doesn't have a static memory address for running. Relocatable code Program code that can be loaded anywhere in memory. Typically the code is divided into *control sections* and all memory addresses are expressed relative to the start of a control section. The compiler/assembler produces a table of all such memory references, and the loader converts them into absolute addresses as part of the loading process. See also position-independent code.A code generated by an assembler or compiler, and in which all memory references needing relocation are either specially marked or relative to the current program-counter reading.

Ans2) Linkers can take objects from a collection called a *library* or runtime library. Most linkers do not include the whole library in the output; they include only the files that are referenced by other object files or libraries. Library linking may thus be an iterative process, with some referenced modules requiring additional modules to be linked, and so on. Libraries exist for diverse purposes, and one or more system libraries are usually linked in by default.

- Reduce the size of your class files (code can be extracted and moved elsewhere where it doesn't disturb anyone).
- Cleaner API since you can't leak internal fields.
- You can test your library independent of your application.
- You can reuse a library in several projects.

Ans3) Programs are not relocatable; they need to be assembled or compiled to execute at specific, predefined addresses. The file contains no relocation or linkage information. These files can be loaded into read/write memory, or stored in read-only memory.DOS COM files are a more recent example of absolute object files. The linker program is used for generating one or more object files and produces an absolute object file with an extension "abs".

Ans4) The main difference between compiler and cross compiler is that the compiler is a software that transforms a computer program written in high-level programming language into machine language while the cross compiler is a type of a compiler that can create an executable code for a platform other than the one on which the compiler is running. A compiler is a translator that is capable of transforming source code into a machine code. There are various types of compilers. The native code compilers convert the source code only for the same type of platform. Incremental compiler compiles the changed lines from the source code and updates the object code. Cross compiler is also a type of compiler. It is capable of creating executable code for various platforms.