**Hardware and Software**

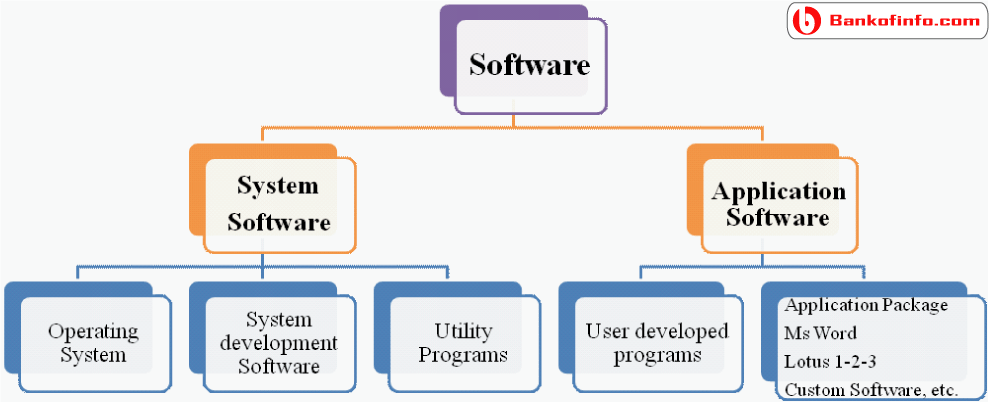
**Hardware** - Physical parts of a computer on which software runs, such as the motherboard, CPU, input, and output devices.

**Software** – all written programs that make a computer function as intended.

Hardware needs instructions from software to perform its function while software need hardware to actually carry out intended tasks.

**Software Classification:**

Software is split into 2 broad categories: application software, and system software – these can be further divided.



**Application Software** – software that performs one or a set of tasks for users. E.g: word processor, browser.

This can in turn be broken down into 3 groups:

General Purpose- Software designed to be widely used in many ways, such as a word processor.

Specialist – software that performs a specific task designed for a specific business, scientific, educational area - such as payroll or air traffic control software.

Bespoke – tailor-made software created to fulfil a specific task for a particular user or organization. It is very expensive but fits the needs of the buyer. E.g: track and trace software.

**System Software** – software needed to control computer hardware and run application software. E.g: OS, antivirus, disk defragmenter.

Operating Systems – a set of programs that manages the operations of the computer. It allows users to use the computer hardware trough application or system software. It’s main roles include presenting a GUI (graphical user interface), managing memory, and managing multi-tasking on the CPU – this involves the use of a scheduler, which ensures all processes get CPU time and the hardware can be used efficiently.

Utility Programs – programs designed to help configure, optimise, or maintain the computer. E.g: antivirus, encryption, backup software, etc.

Libraries – a collection of prewritten programs to allow other programs to run or to be developed. These significantly speed up development, and work across multiple applications.

Translators – Allow programs to be translated into machine code so that programs can be run on a computer.

Interpreters - convert high level languages into machine code one instruction at a time as the program is running.

Compilers - convert high-level languages into machine code before the program is run, the machine code is saved so that the source code can be discarded – this is how software is normally distributed.

Assembler - converts assembly language instructions into machine code. There is a one-to-one correspondence between assembly code instructions and machine code instructions.