

Solutions

Each answer shows the steps, then the **result**.

2.1

- a collection of pre-written routines (subroutines, classes or modules)
- that have already been tested, and that a programmer can include and call in their own program

2.2

- copied into the executable at compile time: **static**
- loaded only when called: **dynamic**
- one shared copy: **dynamic**
- every program must be recompiled: **static**

Statement	static	dynamic
The routines are copied into the executable when the program is compiled.	✓	
The library file is loaded into memory only when the program calls it.		✓
Several running programs share one copy of the library.		✓
Every program that uses the library must be recompiled when the library is updated.	✓	

2.3 any **two** of:

- the executable file is smaller
- the DLL is loaded only when it is needed, so less memory is used
- one copy is shared by several programs
- the DLL can be updated without recompiling the programs that use it

2.4 any **one** of:

- the routine may not do exactly what is needed and cannot be changed
- the program depends on the library being available and correct
- the programmer must learn how the routine is called

2.5 any **two** of:

- the routine is already written, so it saves time
- it has already been tested, so it is reliable
- the programmer does not need to know the mathematics of how to calculate a square root

2.6

- the DLL file must be present (and the correct version) so it can be loaded into memory

3.1

(a) two benefits, each with its consequence:

- the routine is already written and tested, so the team saves the time it would take to write and test path-finding code
- the team does not need expertise in path-finding algorithms, so it can spend its time on the robot's own features
- (also accept) the routine is used by many programs, so it is reliable and the robot is less likely to have a fault in this part

(b) any **one**, described:

- the routine may not handle the robot's exact needs, for example moving obstacles, and the team cannot alter it, so it must work around the limits
- the software depends on the library: a bug or a security weakness in the routine becomes a bug in the robot

3.2

(a) any **two**, each explained:

- the routines are loaded only when the software calls them, so the executable is smaller and memory is used only while the motors are being controlled
- one copy of the DLL is shared by every program that controls the robot, so less memory is used
- the motor routines can be updated or fixed by replacing the DLL, without recompiling the programs that use it

(b) the DLL is loaded at run time, so the older software now runs with the new routines, not the ones it was tested with

- the update changed the behaviour or the interface of a routine, so the calls in the older software no longer work as expected

3.3 two points for each:

- **static:** the game is one larger file that runs on its own, with nothing else to install; the library cannot go missing or be the wrong version, but a graphics bug can only be fixed by shipping a new version of the game
- **dynamic:** the download is smaller and the DLL is shared with other games that use it; the library can be updated separately, but the game will not run if the DLL is missing or the wrong version

3.4

(a) encryption is very hard to write correctly, and a small mistake makes it insecure

- the library routine has been written by experts and tested by many users, so it is far less likely to contain a weakness

(b) the app depends on the library: a weakness or a bug in the library is a weakness in the app

- the programmer did not write the code and may not understand it, so a fault or an out-of-date routine can go unnoticed

3.5

(a) every programmer would call the same routine, so every part of the program checks dates in the same way

- the routine is written and tested once, so the check is correct everywhere, and a fault is fixed in one place
- the programmers save time: nobody writes the check a second time

(b) any **one** of: the library can be reused in the team's next program; a new programmer can use tested routines at once