To show $L \in NP$:
Construct an algorithm A such that, for some polynomials p and q:
(a) for all inputs x, A runs in time $p(|x|)$;
(b) if $x \in L$: then there exists a y such that $A(x, y)$ accepts, where $|y| \leq q(|x|)$;
(c) if $x \notin L$: then for all y, $A(x, y)$ rejects.

y such that $A(x, y)$ accepts is called proof/certificate that $x \in L$.
Often y above is refered to as "guess" of A on input x.
So, A guesses the proof and verifies it.

Reduce problem A to B:

1. Define a polynomial time computable function f.
2. f maps problem instances x of A to problem instances $f(x)$ of B.
3. Show that $x \in A$ iff $f(x) \in B$.

3-SAT problem:

Input: a set V of variables, and a 3-CNF formula.
Question: Is the 3-CNF formula satisfiable?
3-SAT is in NP:
1. Certificates are the truth assignments to the variables. Note that the length of the certificate is proportional to the number of variables, and thus linear in the length of the input.
2. To check a certificate: just evaluate the formula on the truth assignment. It takes time proportional to the length of the formula, and thus linear in the length of the input.

For ease of notation, we usually represent the CNF formula (in SAT or 3-SAT) as set of clauses.
For example,
$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_4 \vee x_5)$.
is represented as the set of two clauses:
$\{(x_1 \vee \neg x_2 \vee x_3), (x_2 \vee \neg x_4 \vee x_5)\}$.

$x_1, \neg x_2$, etc are called literals.

NP Hardness of 3-SAT:
We show $SAT \leq_m^p 3\text{-}SAT$.
Suppose (U, C) is an instance of satisfiability.
We construct an instance (U', C') of 3-SAT such that, C is satisfiable iff C' is satisfiable (and the reduction can be done in poly time).

Suppose $C = \{c_1, c_2, \ldots c_m\}$.
For c_i, we will define C_i' and U_i' below.

$$U' = U \cup \bigcup_{1 \leq i \leq m} U_i'$$

$$C' = \bigcup_{1 \leq i \leq m} C_i'$$

If $c_i = (l_1)$, then

$U'_i = \{y_i^1, y_i^2\}$, and
$C'_i = \{(l_1 \vee y_i^1 \vee y_i^2), (l_1 \vee \neg y_i^1 \vee y_i^2), (l_1 \vee y_i^1 \vee \neg y_i^2), (l_1 \vee \neg y_i^1 \vee \neg y_i^2)\}$,
where y_i^1 and y_i^2 are NEW variables (which are not in U, and not used in any other part of the construction).

If $c_i = (l_1 \vee l_2)$, then

$U'_i = \{y_i^1\}$, and
$C'_i = \{(l_1 \vee l_2 \vee y_i^1), (l_1 \vee l_2 \vee \neg y_i^1)\}$,
where y_i^1 is NEW variable (which is not in U, and not used in any other part of the construction).

If $c_i = (l_1 \vee l_2 \vee l_3)$, then

$U_i' = \emptyset$, and

$C_i' = \{c_i\}$.

If $c_i = (l_1 \vee l_2 \vee \cdots \vee l_r)$, where $r \geq 4$, then

$U_i' = \{y_i^1, \cdots, y_i^{r-3}\}$, and

$C_i' = \{(l_1 \vee l_2 \vee y_i^1),$
$(\neg y_i^1 \vee l_3 \vee y_i^2), \cdots,$
$(\neg y_i^{p-2} \vee l_p \vee y_i^{p-1}), \cdots,$
$(\neg y_i^{r-4} \vee l_{r-2} \vee y_i^{r-3}),$
$(\neg y_i^{r-3} \vee l_{r-1} \vee l_r)\}$,

where $y_i^1, \cdots, y_i^{r-3}$, are NEW variables (which are not in U, and not used in any other part of the construction).

Clearly the transformation can be done in polynomial time.
We claim that C is satisfiable iff C' is satisfiable.

Suppose C is satisfiable.
Fix a satisfying assignment of C. We give a corresponding satisfying assignment of C'.
Variables from U: same truth value as in the satisfying assignment of C.

Other variables are given truth values as follows.
(a) $|c_i| \leq 3$: variables in U_i' are assigned arbitrary truth value (clauses in C_i' are already satisfied).

(b) $|c_i| > 3$:
Suppose $c_i = (l_1, l_2, \cdots, l_r)$. Let l_j be such that l_j is true in the satisfying assignment of C fixed above.

$$(l_1 \vee l_2 \vee y_i^1), (\neg y_i^1 \vee l_3 \vee y_i^2), \cdots$$

$$\cdots (\neg y_i^{j-3} \vee l_j \vee y_i^{j-2}), \cdots$$

$$(\neg y_i^{r-4} \vee l_{r-2} \vee y_i^{r-3}), (\neg y_i^{r-3} \vee l_{r-1} \vee l_r)$$

Then let y_i^k be true for $1 \leq k \leq j-3$, and y_i^k be false for $j-3 < k \leq r-3$. It is easy to verify that all the clauses in C_i' are satisfied.

Now suppose C' is satisfiable. Fix a satisfying assignment of C'.
Then we claim that the truth assignment of U' restricted to U must be a satisfying assignment for C.
To see this suppose $c_i = (l_1 \vee \ldots \vee l_r)$.
(a) $r \leq 3$: then c_i is clearly true due to construction.
(b) $r > 3$:
If y_i^{r-3} is true, then one of l_{r-1}, l_r must be true.
If y_i^1 is false, then one of l_1, l_2 must be true.
Otherwise pick a k such that y_i^k is true but y_i^{k+1} is false. (Note that there must exists such a k). Then l_{k+2} must be true (as we need to satisfy the clause $(\neg y_i^k \vee l_{k+2} \vee y_i^{k+1})$).

Hence C is satisfiable iff C' is satisfiable.
This completes the proof of 3-SAT being NP-complete.

Independent Set Problem:

Input: A graph $G = (V, E)$ and a number k.
Question: Is there a subset $V' \subseteq V$ such that $|V'| = k$ and, for all $v, w \in V'$, $(v, w) \notin E$?
In NP:
– Certificates are subsets V' of V of size k with no edges among them, (clearly of size polynomial in the size of the input).
– Verification: Check that $V' \subseteq V$ and $|V'| = k$, and
for each pair v, w in V' such that $v \neq w$, check if $(v, w) \in E$;
if none of them are in, then accept, Else reject.
Thus, Independent Set Problem is in NP.

NP-Hard:
Reduce 3-SAT to Independent Set.
Given a set X of variables, and a set C of clauses over X, with each clause having exactly 3-literals, form a graph $G = (V, E)$ as follows.
Suppose the i-th clause is $(\ell_i^1 \vee \ell_i^2 \vee \ell_i^3)$.
Let $k = |C|$.
$V = \{u_i, v_i, w_i : 1 \leq i \leq k\}$.
$E = P1 \cup P2$, where
$P1 = \{(u_i, v_i), (u_i, w_i), (v_i, w_i) : 1 \leq i \leq k\}$, and
$P2 = \{(u_i, u_j) : \ell_i^1 = \neg\ell_j^1\} \cup \{(u_i, v_j) : \ell_i^1 = \neg\ell_j^2\} \cup$
$\{(u_i, w_j) : \ell_i^1 = \neg\ell_j^3\} \cup \{(v_i, v_j) : \ell_i^2 = \neg\ell_j^2\} \cup$
$\{(v_i, w_j) : \ell_i^2 = \neg\ell_j^3\} \cup \{(w_i, w_j) : \ell_i^3 = \neg\ell_j^3\}$
Intuitively, u_i represents literal ℓ_i^1, v_i represents literal ℓ_i^2, and w_i represents literal ℓ_i^3.
In P2: "representatives" for literals x_i and $\neg x_i$ are connected.

If the IS (independent set) problem has an independent set of size k, then (X, C) is satisfiable.
Suppose $V' \subseteq V$ is the independent set of size k. Then note that V' must contain exactly one of u_i, v_i, w_i, for each i.
If $u_i \in V'$ then we let l_i^1 to be true.
If $v_i \in V'$ then we let l_i^2 to be true.
If $w_i \in V'$ then we let l_i^3 to be true.
Rest of the variables are assigned arbitrary truth values.
Note that this assignment of truth values makes each clause true.
Also this assignment of truth values is consistent as we do not make both variable x and $\neg x$ true! (as if a vertex representing x is in V', then a vertex representing $\neg x$ cannot be in V' as there will be an edge between these two vertices).

If (X, C) is satisfiable then the IS problem has an independent set of size k.
Suppose (X, C) is satisfiable.
Then each clause has a true literal.
For each i, place in V' one of the vertices from u_i, v_i, w_i such that the literal corresponding to it is true (if several of them are true, choose an arbitrary one).
Now, the set V' is of size k.
It is an independent set as we didn't choose two vertices from any of (u_i, v_i, w_i), thus no edge in $P1$ contains two endpoints in V'. Furthermore, no edge in $P2$ contains two endpoints in V' as these edges only connect vertices corresponding to x and $\neg x$ (for some x) both of which cannot be true.

Clique:

Clique:
Input: A graph $G = (V, E)$ and a number k.
Question: Is there a clique of size k? That is, is there a subset $V' \subseteq V$ such that $|V'| = k$, and for all $v, w \in V'$ such that $v \neq w$, $(v, w) \in E$.
Clique is in NP:
Certificates: Subsets V' of V of size k, where all the edges between pairs of vertices in V' is present in E.
Verification: Given a certificate V', check whether
(a) it is of size k, subset of V and
(b) whether for all $v, w \in V'$ such that $v \neq w$, $(v, w) \in E$.

Reduction: Independent set problem to Clique
If $G = (V, E)$ and k is an Independent set problem, then form a Clique problem as follows:
$G' = (V, E')$, where $E' = \{(v, w) : v \neq w, (v, w) \notin E\}$.
Let $k' = k$

(Intuitively, G' is complement of G)

Then, the Clique problem is (G', k').
It is easy to verify that $V' \subseteq V$ is an independent set in G iff V' is a clique in G'.
Thus, G has an independent set of size k iff G' has a clique of size k.

Hamiltonian Circuit (HC):
Given a graph $G = (V, E)$, Hamiltonian circuit of G is a simple circuit which passes through all the vertices of the graph. That is, a listing $v_1, v_2, \ldots, v_n$ of the vertices of G such that, $v_1, v_2, \ldots, v_n$ are distinct, and $(v_i, v_{i+1}) \in E$, for $1 \leq i < n$, and $(v_n, v_1) \in E$.
It can be shown that Hamiltonian Circuit Problem is NP-complete.

Easy to show that HC $\in$ NP.
Proof of HC is NP-hard is very difficult and beyond the scope of this module.

Travelling Salesman Problem (TSP)
Given a weighted graph $G = (V, E)$, with weight given by wt, and a number w, is there a Hamiltonian circuit in G with weight at most w?
TSP is NP-complete.
Easy to show that TSP is in NP (certificates would be HC which are of weight $\leq w$)

NP-Hardness: Reduction from HC.
Given $G = (V, E)$, a Hamiltonian circuit problem construct a TSP problem $G' = (V', E')$, weight matrix wt and w as follows.
$V' = V$.
$E' = \{(u, v) : u \neq v, u, v \in V\}$
$wt((u, v)) = 1$, if $(u, v) \in E$.
$wt((u, v)) = n + 1$, if $(u, v) \notin E$.
$w = n = |V|$.

Clearly, there are no Hamiltonian circuits in G' of weight $< n$.
It is easy to verify that a Hamiltonian circuit in G is a Hamiltonian circuit in G' with weight n, and a Hamiltonian circuit in G' with weight n is a Hamiltonian circuit in G.
Thus, $HC \leq_m^p TSP$.
Hence TSP is NP-complete.