1. Not doing question 1.

2. (a) The formula for a general linear multistep method of order k is:

$$\sum_{i=0}^{l} \alpha_i y_{n+i} = h \sum_{i=0}^{k} \beta_i f_{n+i}$$

The method is explicit if $\beta_k = 0$, and implicit if $\beta_k \neq 0$. If the method is implicit, then the value $y_n + k$ must be computed using:

$$y_{n+k} = h\beta_k f(x_{n+k}, y_{n+k}) + h \sum_{i=0}^{k-1} \beta_i f_{n+i} - \sum_{i=0}^{k-1} \alpha_i y_{n+i}$$

(b) In order to determine the order of convergence of a linear multistep method, we must determine the constants $C_0, \ldots, C_j$:

$$C_0 = \sum_{i=0}^{k} \alpha_i$$
$$C_1 = \sum_{i=1}^{k} i\alpha_i - \sum_{i=0}^{k} \beta_i$$
$$C_j = \frac{1}{j!} \sum_{i=1}^{k} i^j \alpha_i - \frac{1}{(j-1)!} \sum_{i=1}^{k} j^{i-1} \beta_i$$

The order of the method p is given by $C_0 = ... = C_p = 0, C_{p+1} \neq 0$.
For consistency, p must be greater than or equal to 1.
For stability, the roots of the first characteristic polynomial must have modulus less than or equal to one and any root that does have a modulus equal to one must be simple. The first characteristic polynomial is given by:

$$p(\zeta) = \sum_{i=0}^{k} \alpha_i \zeta^i$$

(c) i. In order to determine the order of method two, we must find the values of α and β and the constants:

$$\alpha_0 = 0$$
$$\alpha_1 = \frac{1}{6}$$
$$\alpha_2 = -1$$
$$\alpha_3 = \frac{1}{2}$$
$$\alpha_4 = \frac{1}{3}$$
$$\beta_0 = 0$$
$$\beta_1 = 0$$
$$\beta_2 = 1$$

Now we can find the constants:

$$C_0 = \sum_{i=0}^{k} \alpha_i = \frac{1}{6} - 1 + \frac{1}{2} + \frac{1}{3} = 0$$
$$C_1 = \sum_{i=1}^{k} i\alpha_i - \sum_{i=0}^{k} \beta_i = \frac{1}{6} - 2(-1) + \frac{3}{2} + \frac{3}{3} - 1 = -\frac{1}{3}$$

Therefore we can see that the method is not consistent as it is of order 0.
The first characteristic polynomial is:

$$p(\zeta) = \frac{1}{3}\zeta^3 + \frac{1}{2}\zeta^2 - \zeta + \frac{1}{6}$$

Since we can (relatively easily) see that $\zeta = 1$ is a root, we can use polynomial division to get the rest:

$$\begin{array}{r|l} & \frac{1}{3}x^2 + \frac{5}{6}x - \frac{1}{6} \\ \hline x - 1 & \frac{1}{3}x^3 + \frac{1}{2}x^2 \quad - x + \frac{1}{6} \\ & \underline{-\frac{1}{3}x^3 + \frac{1}{3}x^2} \\ & \quad \frac{5}{6}x^2 \quad - x \\ & \quad \underline{-\frac{5}{6}x^2 + \frac{5}{6}x} \\ & \quad\quad -\frac{1}{6}x + \frac{1}{6} \\ & \quad\quad \underline{\frac{1}{6}x - \frac{1}{6}} \\ & \quad\quad\quad 0 \end{array}$$

Now we have $(\zeta - 1)\left(\frac{\zeta^2}{3} + \frac{5\zeta}{6} - \frac{1}{6}\right)$, we can use the quadratic formula to find the rest of the roots:

$$a = \frac{1}{3}, b = \frac{5}{6}, c = \frac{-1}{6}$$
$$\zeta = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-5 \pm \sqrt{33}}{4}$$
$$\zeta = 2.686, \zeta = -0.1864$$

Since 2.686 is greater than 1, the method is not stable.

ii. Ouch...

iii. The results differ as the method is not stable. Instead, an implicit method should be used that is stable.

3. Not doing this one.

4. (a)
- Selection and variation are the main operators.
- Selection ensures that members of the population that have a low fitness are removed, and ones with a high fitness are retained. Fitness is determined by the value of the objective function when it is passed in the candidate solution. Selection tries to direct the population towards 'better' areas of the parameter space by removing solutions in 'bad' areas of the paramters space.
- Variation ensures that different individuals in the population of candidate solutions are not the same. It often works by generating completely new individuals, or mutating/combining existing individuals. This is important as it is how new 'ideas' are added to the population.
- Selection can be achived by a variety of different means, such as:
  - Stochastic ranking; rank the members of the population in terms of fitness, but have a small random chance of getting the ranking wrong. Select the top n members, but discard the others.

  - Tournament: Compare the fitness of each individual to m other individuals ($m << N$). Rank the individuals by how many they won in the tournament and take the top n.
  - Roulette: Arrange the individuals around a roulette wheel, and make the size of their portion of the wheel proportional to their fitness. Spin the wheel and whoever it lands on remains in the population (repeat until n remaining).
- Variation can also be achieved through different means:
  - Most algorithms implement some form of mutation, where existing solutions are changed by some random peturbation. Examples include bit flipping of binary genes, adding a random number (specified by a standard deviation gene) to floating point genes, and acceleration vectors in particle swarm.
  - Some genetic algorithms sexually combine candidate solutions, which can be achieved by cross over (trees, binary genes), averaging floating point genes etc.
- Evolutionary algorithms are efficient at escaping local minima because:
  - Since they have a population of candidate solutions rather than just one, there is little chance that all of the candidate solutions will be in the same part of the parameter space (and hence potentially residing in a local minima). Instead, there is more chance that different groups within the population will gravitate towards different minima, some of which may be local, but others could be global.
  - Variation tries to ensure that candidate solutions are distributed well over the parameter space, so all the solutions getting stuck in a local minima is unlikely. Furthermore, variation can also force candidates out of a local minima through mutation.
  - Sexually reproducing solutions can cause their offspring to occupy a completely different area of the search space.

(b) **Binary genes** are mutated by flipping each bit in the binary string with a probability p. p is usually very low to avoid mutating more than just one or two bits.

Floating point genes have a companion standard deviation gene. This is mutated before the actual gene is mutated, and then used to derive a random value that is added to the gene itself. In this manner, the rate at which the gene mutates changes as time goes on. This was invented by Rechenberger and Schwefel.

Parse trees are mutated by replacing a sub-tree with a randomly generated sub-tree. In order to avoid bloat, properties may be required of the randomly generated sub-tree, such as it being the same (or smaller) depth as the removed sub-tree, or it having less than some number of nodes.

(c) The differences between GP and other EA's include:

- GP suffers from bloat, where the size of the solution grows, but the fitness of the solution remains constant. This can be remidied in a mnumber of ways, for example the fitness of large trees could be reduced. Most other EA's do not suffer from bloat.
- Genetic programming represents each candidate solution as a tree, while most other EA's represent solutions as vectors (e.g. particle swarm), integers (e.g. binary genes) etc
- Rather than producing a candidate solution such as a set of parameters to minimise the objective function, genetic programming produces a function that fits a specific purpose. Ultimately, these two things are the same, since functions can be mapped onto the set of integers just like the candidate solutions of other EA's can, but GA's operate at a higher level of abstraction, and from a different perspective.
- Following on from this, GA's can often encapsulate control structures such as loops, if statements etc. Traditional EA's do not do this.

5. (a) A scale free model best captures the structure of actors in Hollywood. The general properties of a scale free model are:

- A power law degree distribution (lots of nodes with a very low degree, a minority of nodes with a very high degree). Some nodes are hub nodes with lots of incident edges.

- A high clustering efficient is typical of scale free networks. This is to be expected as the clustering coeffieicnt is defined as being the number of triangles connected to a node, divided by the number of triples centered on the node. As such, the network is viewed as dense, since it is so well connected.
- The average path length of the network and the diameter is small in comparison to the size of the network, since there is usually a route through one or more hubs between any node and any other node.

In the context of Hollywood, some actors are very well connected in the sense that they have been in a lot of popular films (often with big budgets and large casts) and as such have a high degree. At the same time, there are many actors that have acted in only one or two films (maybe they weren't very good, or only play background roles and aren't well known by directors).

Obviously the network is large, since there are many many actors in the world. The size of the network is an important criteria for it to be considered complex, and scale free.

Furthermore, scale free networks encapsulate the fact that actors with large numbers of edges will likely acquire more edges as they feature in more popular films. Some models of scale free networks also model the growth of the network, which is an integral part of real world networks.

(b) The Barabasi-Albert (1999) model is able to create scale-free networks. Start with a small set of nodes, and then for each time step:

- Add a new node to the graph with m links.
- The probability distribution for what other nodes to connect the new node to is:

$$\Pi(k_i) = \frac{k_i}{\sum_{j=0}^{N} k_j}$$

(c) In 2000, the BA model was expanded to include re-wiring, and in 2001, it was again update to include competition:

Re-wiring: Each edge would be re-wired with a probability p, and the node it would connect to would be chosen in a similar manner to in the original model, except it would allow isolated nodes to be picked:

$$\Pi(k_i) = \frac{k+1}{\sum_{j=0}^{k} (k_j + 1)}$$

In the re-wiring model, we start with isolated nodes and for each time step ($0 \to T$), we do the following:

- Add m links with a probability p to each node using the distribution given above.
- With a probability q, rewire m links using the same distribution Π.
- Finally, add a new node with m links with a probability $1 - q - p$, and use Π to choose which other nodes to connect it to.

Competition: Competition encapsulates the fact that all not all nodes are the same; real world networks are heterogenous. For example, some actors are better than others, and even though one actor may have been in lots of films, another actor may be a young star, and they might feature in a string of films in quick succession. This is modelled by giving each node a parameter μ when it is added to the network.

At the start of this algorithm, the nodes are isolated (as with re-wiring), and for each time step, a new node is added with a fitness $0 < \mu < 1$ taken from a uniform probability distribution, and m new links are added to the new node using the preferential attachment distribution:

$$\Pi(k_i) = \frac{k_i \mu_i}{\sum_{j=0}^{k} (k_j + 1)\mu_j}$$

6. Didn't do this one.