1. Part A and B
   1. A

| public DownloadInfo getDownloadInfo (String title) {  downloadInfo reference = null;  for(downloadInfo downloaded : downLoadList) {  if(downloaded.getTitle().equals(title)) {  reference = title;  }  }  return reference;  } |
| --- |

* 1. B

| public void updateDownloads(List<String> titles) {  for( String title : titles) {  if (getDownloadInfo(title) != null) {  getDownloadInfo(title).incrementTimesDownloaded();  }  else{  downloadList.add(new DownloadInfo(title));  }  }  } |
| --- |

1. Parts A and B
   1. A

| public TokenPass (int playerCount) {  board = new int[playerCount];  for(int i = 0; i < board.length; i ++){  board[i] = (int)(math.random() \* 10) + 1;  }  currentPlayer = (int)(math.random() \* (playerCount - 1));  } |
| --- |

* 1. B

| public void distributeCurrentPlayerTokens() {  int count = 0;  currentPlayerPosition = board[currentPlayer];  board[currentPlayer] = 0;  while(count <= currentPlayerPosition) {  for(int j = 0; j <= board.length; j++) {  board[j] += 1;  count++;  }  }  } |
| --- |