

Mr. Giansante



C++ Programming
Hangman

August 2018

Hangman

Design a C++ program that simulates a one-player game of Hangman.

Despite its grim-sounding name, Hangman is a traditional and enjoyable game.

Playing is simple. You are presented with a picture of a gallow and a string of dashes.

The dashes represent the letters of the word (or words) that you have to guess.

You must guess one letter at a time. If you choose a letter that is contained within the word, a dash will be replaced with that letter, giving you a better chance of working out the word.

If you guess a letter not within the word, one bodypart will appear on the HangMan on the gallows. Bodyparts usually are (in order that they appear): head, body, left arm, right arm, left leg, right leg.

You must guess the correct word before the hangman is completed.

Hangman Help

Enter your puzzles in an array (of strings)

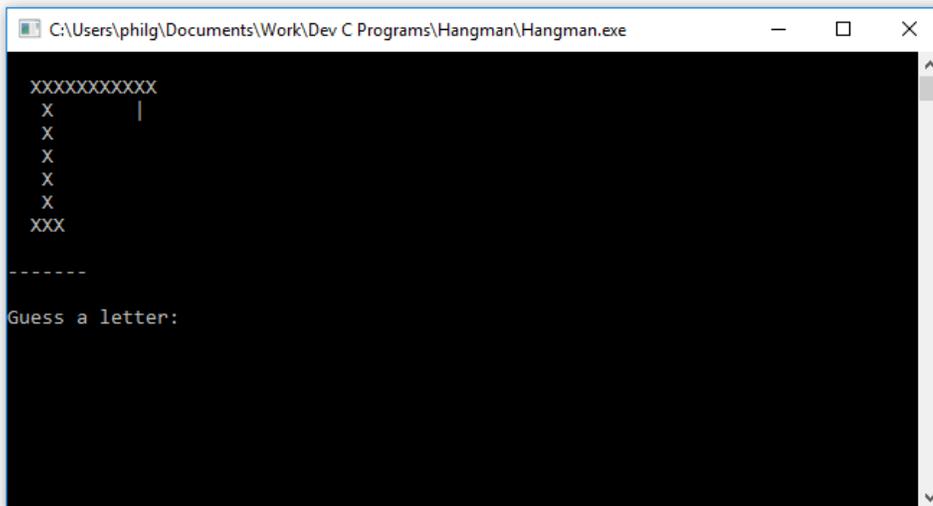
```
puzzles[1] = "The Empire Strikes Back";  
puzzles[2] = "Casablanca";  
puzzles[3] = "This is Spinal Tap";
```

Write a function that will take all the letters in the selected puzzle and replace them with dashes (-). Characters that are not letters (ie. numbers, apostrophe's) should not be converted to dashes.

Example. The Empire Strikes Back
 becomes ...

Example. Who's that Girl?
 becomes ...

---'s ----?---



Hangman

Libraries to Include

```
#include <iostream>
#include <ctime>      // time()
#include <stdlib.h>   // srand(), rand()
```

Variables

```
string puzzles[150];
string currentpuzzle;
string puzzledashes;
string newdashes;
char guess;
int r;
int strikes;
bool gameover = false;

puzzles[1] = "The Empire Strikes Back";
puzzles[2] = "Casablanca";
puzzles[3] = "This is Spinal Tap";

etc.
```

Picking a Random Puzzle

```
srand(time(NULL));

do
{
    r = rand()%100 + 1;
    currentpuzzle = puzzles[r];
    // Begin by converting all letters in puzzle to uppercase
    for(int i = 0; i < currentpuzzle.length(); i++)
    {
        currentpuzzle.at(i) = toupper(currentpuzzle.at(i));
    }
    puzzledashes = makedashes(currentpuzzle);
```

 Note: makedashes() is a function that you will need to create. See below.

Creating the Dashes

The makedashes() function converts all the letters in the puzzle to dashes. It ignores numbers and punctuation. The parameter is the original puzzle that was randomly chosen.

The function goes through the puzzle character-by-character using a for loop. Recall that the first character of a string is at position 0 and the last one at position (length of string - 1).

Hangman

```
string makedashes(string puzzletconvert)
{
    string temp = "";
    // Scan puzzle character by character and replace letters (ie. alpha) with dashes
    // All other characters (ie. - ' !) and numbers should be kept as is
    for(int i = 0; i < puzzletconvert.length(); i++)
    {
        if(isalpha(puzzletconvert.at(i)) == true)
        {
            temp = temp + "-";
        }
        else
        {
            temp = temp + puzzletconvert.at(i);
        }
    }
    return temp;
}
```

The searchforletter() Function

The searchforletter() function searches for letter that the user guessed to see if it is in the puzzle.

The function goes through the puzzle character-by-character using a for loop. Recall that the first character of a string is at position 0 and the last one at position (length of string - 1).

```
string searchforletter(string searchin, string newdashes, char searchfor)
{
    for(int i = 0; i < searchin.length(); i++)
    {
        if(searchin.at(i) == searchfor)
        {
            newdashes.at(i) = searchfor;
        }
    }
    return newdashes;
}
```

Displaying the Hang Man

The drawhangman() function will draw the hangman using ASCII graphics.

You must create this function yourself.

The parameter that you should pass the function is a number representing the number of strikes that the player has.

```
XXXXXXXXXXXXX
X             |
X             O
X            /|\
X           / \
X
XXX
```

```
void drawhangman(int num)
{
    // Place your code here
}
```

Hangman



Learning Outcomes

Programming Concepts and Skills

- A1. demonstrate the ability to use different data types, including one-dimensional arrays, in computer programs
- A2. demonstrate the ability to use control structures and simple algorithms in computer programs
- A3. demonstrate the ability to use subprograms within computer programs
- A4. use proper code maintenance techniques and conventions when creating computer programs

Software Development

- B1. use a variety of problem-solving strategies to solve different types of problems independently and as part of a team
- B2. design software solutions to meet a variety of challenges
- B3. design algorithms according to specifications
- B4. apply a software development life-cycle model to a software development project

Please fill out all the information in this column using a pen before getting this assignment marked.

Name

_____ _____
Date Class

Academic Honesty

The work I am submitting is completely my own creation and has not been copied from anyone else's work. If I have received help on this project, the names of those who have assisted are listed below.

Signature

Pre-Marking

The following people have pre-marked this assignment:
(minimum of two)

User Interface

- Appropriate Title, Author Information, etc.
- Efficient and Esthetically-Pleasing User Interface
- Spelling and Grammar are correct

Code / Programming Style

- Variables are declared and logically named
- Code is commented where appropriate
- Code is indented and spaced to show structure
- Code is efficient

Work Ethic / Problem Solving Skills

- Problem Solving Skills are demonstrated
- Makes Productive Use of Time
- Only Seeks Help when Necessary

Program-Specific Criteria

- Puzzle appears at start up
- Puzzles are picked randomly
- Dashes are determined properly
- Numbers or punctuation in puzzles are dealt with
- Letters are placed correctly to replace dashes
- Part of hangman is added when incorrect guess
- Answer is shown if player wins or loses
- User can play another game or quit

Teacher Comments

- Level 1 - Limited ability to meet standard and limited effectiveness
- Level 2 - Some ability demonstrated and moderately effective
- Level 3 - Considerable ability demonstrated, considerable clarity or accuracy
- Level 4 - Thorough, high degree of skill demonstrated, insightful, highly accurate

Level