1. **Code Segment:**

#include <iostream>

#include <cstdlib>

#include<ctime>

#include <string>

#include <bits/stdc++.h>

using namespace std;

const int MAX\_TRY=5

class WELCOME

{

int choice;

public:

// constructor to welcome the player

WELCOME()

{

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WELCOME TO THE HANGMAN GAME \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

cout << "\nRules of THE HANGMAN GAME are listed below\n\n";

cout << "\tFirstly, you need to pick the category of the word.\n";

cout << "\tAccording to the category you chose, word will be displayed.\n";

cout << "\tWord will be displayed secretly in the form of the star.\n";

cout << "\tYou can type only one letter in one try.\n";

cout << "\tMaximum " << MAX\_TRY << " times you can do mistake.\n";

cout << "\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

}

// function to let player choose the category of word to play

int choose()

{

cout<<"\n1.ANIMAL\t2.COUNTRY\t3.PUBLIC FIGURE"<<endl;

cout<<"Enter your choice\n";

cin>>choice;

return choice;

}

};

class UPPER

{

string str;

public:

// funtion to convert string to uppercase

uppers(string str)

{

transform(str.begin(), str.end(), str.begin(), ::toupper);

cout << str << endl;

}

};

class THE\_GAME : public UPPER

{

char yn;

public:

/\* Take a one character guess and the secret word, and fill in the

unfinished guessword. Returns number of characters matched.

Also, returns zero if the character is already guessed. \*/

int fill\_letter (char guess, string secretword, string &guessword)

{

int i;

int matches=0;

int len=secretword.length();

// loop to return no of matches if guess is correct

for (i = 0; i< len; i++)

{

// Return 0 if this letter is matched already in a previous guess

if (std :: toupper(guess) == std :: toupper(guessword[i]))

return 0;

// increases the matched if the guess is in the secret word

if (std :: toupper(guess) == std :: toupper(secretword[i]))

{

guessword[i]= std :: tolower(guess);

matches++;

}

}

return matches;

}

// type bool funtion to return true if player wants to play again and false if doesn't

bool playagain()

{

cout << "\nDo you wanna play again? (Y/N) : ";

cin>>yn;

if ((yn == 'Y') || (yn == 'y'))

return true;

else

return false;

}

};

int main ()

{

int choose;

WELCOME wel;

THE\_GAME game;

// againplay label after player wants to play the game again

againplay:

// animals string array

string animals[] =

{

"panther",

"lion",

"tiger",

"cheetah",

"elephant",

"leopard",

"hyena",

"giraffe",

"goat",

"dinosaur",

"hippopotamus",

"chimpanzee",

"buffalo",

"rhinoceros",

"camel",

"blackbuck",

"panda",

"whale",

"python",

"crocodile"

};

// countries string array

string countries[] =

{

"nepal",

"india",

"bangladesh",

"afghanistan",

"australia",

"malaysia",

"portugal",

"mexico",

"america",

"indonesia",

"spain",

"switzerland",

"germany",

"netherlands",

"belgium",

"slovakia",

"singapore",

"bhutan",

"kuwait",

"taiwan"

};

// public\_figures string array

string public\_figures[] =

{

"obama",

"salman",

"narendra",

"leonardo",

"hawking",

"mahendra",

"sachin",

"dravid",

"nagarjuna",

"prabhash",

"katrina",

"shakira",

"neymar",

"ronaldo",

"mahesh",

"jayalalithaa",

"amitabh",

"hrithik",

"rajinikanth",

"deepika"

};

string name;

char letter;

int count\_wrong\_guess=0;

string word;

//choose and copy a word from array of words randomly

srand(time(NULL));

int n=rand()% 10;

again:

choose = wel.choose();

switch (choose)

{

case 1 : word = animals[n];

break;

case 2 : word = countries[n];

break;

case 3 : word = public\_figures[n];

break;

default : cout << "Invalid choice";

goto again;

}

// Initialize the secret word with the \_ character.

string unknown(word.length(),'\*');

cout << "\nIts a " << unknown.length() << " letters word";

cout << "\nFind the word.\nAll the best...";

// Loop until the guesses are used up

while (count\_wrong\_guess < MAX\_TRY)

{

cout << "\n\n" << unknown;

cout << "\n\nGuess a letter: ";

cin >> letter;

// Fill secret word with letter if the guess is correct,

// otherwise increment the number of wrong guesses.

if (game.fill\_letter(letter, word, unknown)==0)

{

cout << endl << "Whoops! That letter isn't in there!" << endl;

count\_wrong\_guess++;

// Tell player how many guesses has left now

cout << "You have now " << MAX\_TRY - count\_wrong\_guess << " guesses left." << endl;

}

else

{

cout << endl << "You found a letter!" << endl;

// Tell player how many guesses has left still

cout << "You have still " << MAX\_TRY - count\_wrong\_guess << " guesses left." << endl;

}

// diplay player that he/she won the game

if (word==unknown)

{

cout << "\nCONGRATULATION YOU WON THE HANGMAN GAME\n";

cout << "\nYou found the correct word ::\t";

game.uppers(word);

break;

}

}

// display player that he/she lost the game

if(count\_wrong\_guess == MAX\_TRY)

{

cout << "\nSorry, you lose..... You are HANGED." << endl;

cout << "The word was ::\t";

game.uppers(word);

}

//cin.ignore();

//cin.get();

if(game.playagain() == true)

goto againplay;

else

{

cout << "\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n";

cout << "Thank you for playing THE HANGMAN GAME with us\nSee you agian...........";

cout << "\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n";

exit(0);

}

}