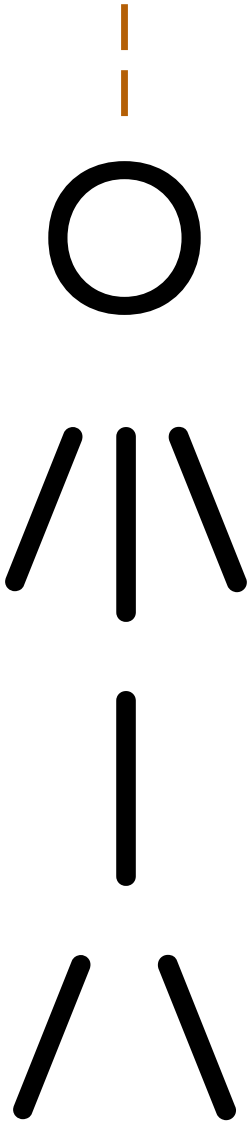
Project 2

[Hangman]



CIS-5-42376

Name: Blake Ward

Date: 04/21/18

Intro

Title: Hangman

This is a single player only version of the game “Hangman”, where you will be given a random word to try and figure out what the hidden word is.

Instructions

Step 1: Have the student select a letter of the alphabet.

Step 2: If the letter is contained in the word, the computer program will have the letter you guessed show up on the letter grid while eliminating the letter you choose from being used again.

Step 3: If the letter is not contained in the word, then a part of the Hangman will be drawn on screen. If you continue to type in the wrong characters then more of the Hangman will be drawn until you get the full Hangman, and when that happens that means it's “Game Over”. It starts off with two ropes, then the head, then the arms and body, and lastly the legs.

Step 4: The game continues until the word is guessed (all letters are revealed). If you guess all the characters correctly, you win! If all the parts of the hangman are displayed, then that means you have lost.

Summary

```
Project 1 size: 233 Lines Total
Project 2 size: 272 Lines main.cpp,
                38 Lines Player.cpp,
                51 Lines Player.h,
                361 Lines Total
The numbers of variables: About 30
```

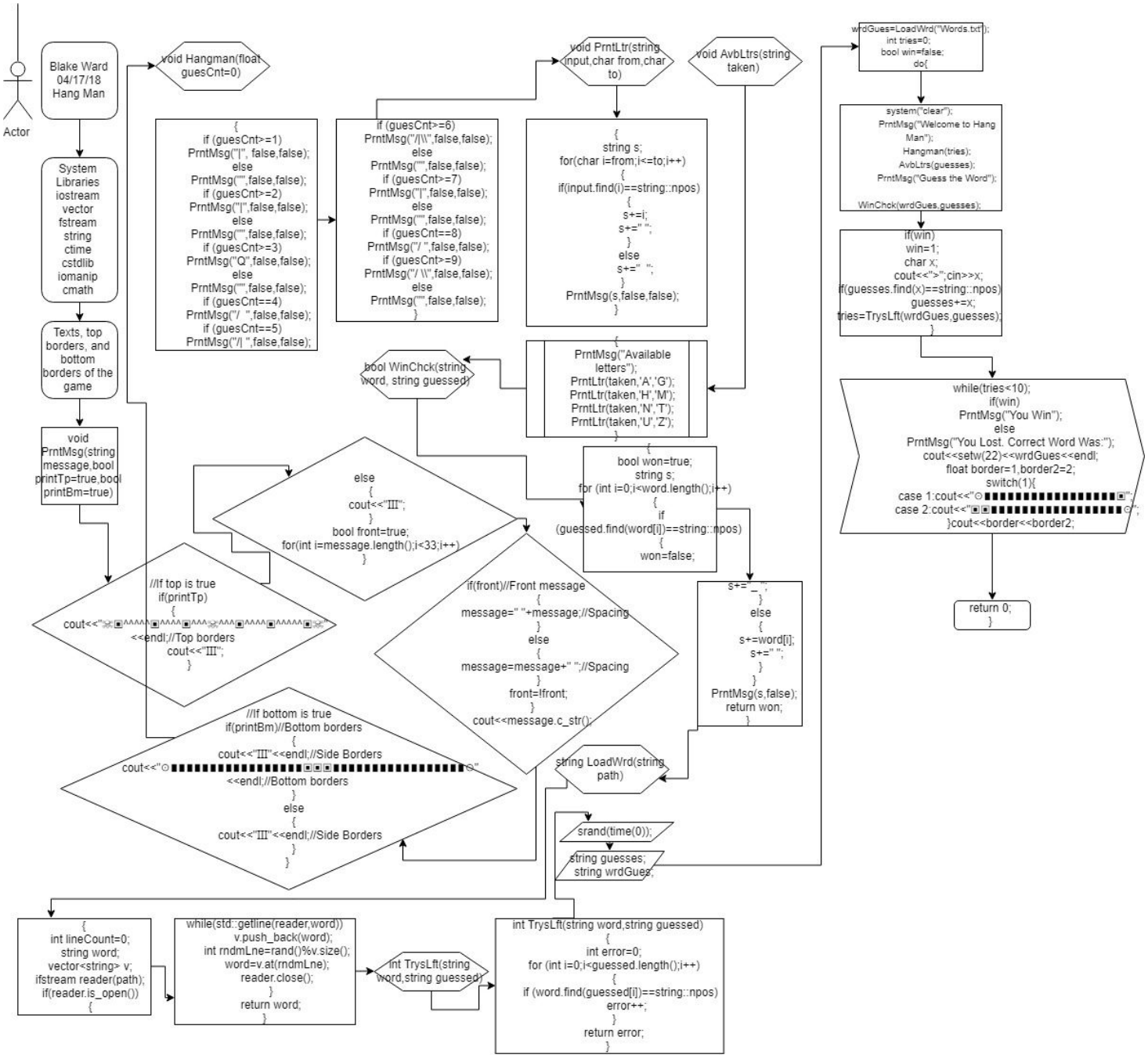
I have never done a project like this before and it's one of the hardest projects that I have ever done. Even though it's most likely nowhere near as difficult as what some game developers do now at days, I'm glad I got to get a taste of what process they go through to make a game.

This took me about a week and a half to do and there were definitely a few problems I had to keep trying to figure why they were not working so that I could make the whole program work, but it was alright after I finally solved them. I also had to go a few chapters ahead to find some new libraries to make my game work properly.

Reference

1. Gaddis & Savitch Textbook
2. Hangman wikipedia (https://en.wikipedia.org/wiki/Hangman_(game))

Flow Chart


Actor
Blake Ward
04/17/18
Hang Man
System
Libraries
iostream
vector
fstream
string
ctime
cstdlib
iomanip
cmath
Texts, top
borders, and
bottom
borders of the
game
void
PrntMsg(string
message,bool
printTp=true,bool
printBm=true)
void Hangman(float
guesCnt=0)
{
if (guesCnt>=1)
PrntMsg("|", false,false);
else
PrntMsg("",false,false);
if (guesCnt>=2)
PrntMsg("|",false,false);
else
PrntMsg("",false,false);
if (guesCnt>=3)
PrntMsg("Q",false,false);
else
PrntMsg("",false,false);
if (guesCnt==4)
PrntMsg("/ ",false,false);
if (guesCnt==5)
PrntMsg("/| ",false,false);
if (guesCnt>=6)
PrntMsg("/|\",false,false);
else
PrntMsg("",false,false);
if (guesCnt>=7)
PrntMsg("|",false,false);
else
PrntMsg("",false,false);
if (guesCnt==8)
PrntMsg("/ ",false,false);
if (guesCnt>=9)
PrntMsg("/ \",false,false);
else
PrntMsg("",false,false);
}
void PrntLtr(string
input,char from,char
to)
{
string s;
for(char i=from;i<=to;i++)
{
if(input.find(i)==string::npos)
{
s+=i;
s+=" ";
}
else
s+=" ";
}
PrntMsg(s,false,false);
}
void AvbLtrs(string
taken)
{
PrntMsg("Available
letters");
PrntLtr(taken,'A','G');
PrntLtr(taken,'H','M');
PrntLtr(taken,'N','T');
PrntLtr(taken,'U','Z');
}
bool WinChck(string
word, string guessed)
{
bool won=true;
string s;
for (int i=0;i<word.length();i++)
{
if
(guessed.find(word[i])==string::npos)
{
won=false;
s+="_";
}
else
{
s+=word[i];
s+=" ";
}
}
PrntMsg(s,false);
return won;
}
wrdGues=LoadWrd("Words.txt");
int tries=0;
bool win=false;
do{
system("clear");
PrntMsg("Welcome to Hang
Man");
Hangman(tries);
AvbLtrs(guesses);
PrntMsg("Guess the Word");
WinChck(wrdGues,guesses);
if(win)
win=1;
char x;
cout<<">";cin>>x;
if(guesses.find(x)==string::npos)
guesses+=x;
tries=TrysLft(wrdGues,guesses);
}
while(tries<10);
if(win)
PrntMsg("You Win");
else
PrntMsg("You Lost. Correct Word Was:");
cout<<setw(22)<<wrdGues<<endl;
float border=1,border2=2;
switch(1){
case 1:cout<<"⊙▬▬▬▬▬▬▬▬▬▬▬▬▬▣";
case 2:cout<<"▣▣▬▬▬▬▬▬▬▬▬▬▬▬⊙";
}cout<<border<<border2;
return 0;
}
else
{
cout<<"III";
}
bool front=true;
for(int i=message.length();i<33;i++)
{
//If top is true
if(printTp)
{
cout<<"⊙▣^^^^^▣^^^^▣^^^⊙^^^▣^^^^▣^^^^▣⊙"
<<endl;//Top borders
cout<<"III";
}
if(front)//Front message
{
message=" "+message;//Spacing
}
else
{
message=message+" ";//Spacing
}
front=!front;
}
cout<<message.c_str();
//If bottom is true
if(printBm)//Bottom borders
{
cout<<"III"<<endl;//Side Borders
cout<<"⊙▬▬▬▬▬▬▬▬▬▬▬▣▣▣▬▬▬▬▬▬▬▬▬▬▬▬⊙"
<<endl;//Bottom borders
}
else
{
cout<<"III"<<endl;//Side Borders
}
}
string LoadWrd(string
path)
{
int lineCount=0;
string word;
vector<string> v;
ifstream reader(path);
if(reader.is_open())
{
while(std::getline(reader,word))
v.push_back(word);
int rndmLne=rand()%v.size();
word=v.at(rndmLne);
reader.close();
}
return word;
}
srand(time(0));
string guesses;
string wrdGues;
int TrysLft(string
word,string guessed)
int TrysLft(string word,string guessed)
{
int error=0;
for (int i=0;i<guessed.length();i++)
{
if (word.find(guessed[i])==string::npos)
error++;
}
return error;
}

Cross-List

Project 1 Cross list:

Chapter	Section	Topic	Where Line #'s	Pts	Notes
2	2	Cout	26-27, 31, 46, 51-52, 56, 224-225		
	3	Libraries	9-16	8	iostream, iomanip, cmath, cstdlib, fstream, string, ctim
	4	Variables/literals	20, 61, 104, 121		No variables in global area, failed project!
	5	Identifiers	34, 107		
	6	Integers	34, 156, 164, 172-175, 192, 202	3	
	7	Characters	104, 107, 192, 214	3	
	8	Strings	20, 104, 106, 109, 121, 131, 134, 137, 153, 156, 157, 172, 178, 197-198, 216	3	
	9	Floats no doubles	61	3	Using doubles will fail the project, floats OK!
	10	Bools	20, 33, 131, 133, 203	4	
	11	Sizeof*****			
	12	Variables 7 char. or less			
	13	Scope ***** no global variables			All variables <= 7 characters

	14	Arithmetic operators			
	15	Comments 20%+		5	Model as pseudo code
	16	Named constants			All Local, only Conversions/Physics/Math in Global area
	17	Programming style ***** Emulate			Emulate style in book/in class repository

Chapter	Section	Topic	Where Line #'s	Pts	Notes
3	1	Cin	215		
	2	Math Expression			
	3	Mixing data types *****			
	4	Overflow/Underflow ****			
	5	Type Casting	24-46, 49-58, 62-101, 104-118, 123-128, 131-150, 154-169, 173-182, 204-225	4	
	6	Multiple assignment *****			
	7	Formatting output	200-225	4	
	8	Strings	20, 104, 106, 109, 121, 131, 134, 137, 153,	3	

			156, 157, 172, 178, 197-198, 216		
	9	Math Library	16	4	All libraries included have to be used
	10	Hand tracing ******			

Chapter	Section	Topic	Where Line #'s	Pts	Notes
4	1	Relational Operators			
	2	if	160,178	4	Independent if
	4	If-else	24-29, 36-40, 49-54, 109-114, 137-142, 220-222	4	
	5	Nesting	107, 135, 175	4	
	6	If-else-if	63-99	4	
	7	Flags *****			
	8	Logical operators	34-44, 61-97, 107-115, 133-145, 155, 164-165, 174-179, 201-203, 213-219	4	
	11	Validating user input	215	4	
	13	Conditional Operator	20, 24, 34, 36, 49, 61-100, 104, 107-109, 117, 121-127, 131,	4	

			135-137, 148, 153, 159-165, 172, 175-178, 192-194, 201, 206-212, 216-224		
	14	Switch	226	4	

Chapter	Section	Topic	Where Line #'s	Pts	Notes
5	1	Increment/Decrement	34, 107, 135, 175, 179	4	
	2	While	162, 216	4	
	5	Do-while	204	4	
	6	For loop	107, 135, 175	4	
	11	Files input/output both	201-225	8	
	12	No breaks in loops *****			Failed Project if included

Cross List For Project 2

Chapter	Section	Topic	Where Line #'s	Pts	Notes
13		Classes			
	1 to 3	Instance of a Class	17	4	
	4	Private Data Members	44	4	Never Public

	5	Specification vs. Implementation	18, 44	4	.h vs. .cpp files Always split
	6	Files input/output both	208, 209, 223, 224, 228-230, 240	4	
	7, 8, 10	Constructors	17	4	Overloading
	9	Destructors	26	4	
	12	Arrays of Objects	140, 181	4	
	16	UML		4	

Chapter	Section	Topic	Where Line #'s	Pts	Notes
14		More about Classes			
	1	Static		5	
	2	Friends		2	
	4	Copy Constructors	22	5	
	5	Operator Constructors	21	8	Overload 3 operators
	7	Aggregation	29	6	

Chapter	Section	Topic	Where Line #'s	Pts	Notes
15		Inheritance			
	1	Protected members	44	6	

	2 to 5	Base Class to Derived		6	
	6	Polymorphic associations	14-51	6	
	7	Abstract Classes		6	

Chapter	Section	Topic	Where Line #'s	Pts	Notes
16		Advanced Classes			
	1	Exceptions		6	
	2 to 4	Templates		6	
	5	STL	9, 10	6	
		Sum		100	

Program

main.cpp:

```
/*
 * File:   main.cpp
 * Author: Blake Ward
 * Created on April 17, 2018, 7:30 PM
 * Purpose:  A game of Hang Man.
 */

//System Libraries
#include <iostream> //I/O Library -> cout,endl
#include <vector>   //Initializing its Contents
#include <fstream>  //File i/o
#include <string>   //String
#include <ctime>    //Time
#include <cstdlib>  //Rand/Srand
#include <iomanip>  //Format Library
#include <cmath>    //Mathematical Operations

#include "Player.h"

using namespace std;//namespace I/O stream library created

//Texts, top borders, and bottom borders of the game
void PrntMsg(string message,bool printTp=true,bool printBm=true)

   //If top is true
{
    if(printTp)//Top borders
   {
      cout<<"☠▣^^^^^▣^^^^▣^^^☠^^^▣^^^^▣^^^^^▣☠"<<endl;//Top borders
      cout<<"Ⅲ";//Side Borders
   }
   else
   {
      cout<<"Ⅲ";//Side Borders
```

```
    }
    bool front=true;
    for(int i=message.length();i<33;i++)//Length of borders
    {
       if(front)//Front message
       {
          message=" "+message;//Spacing
       }
       else
       {
          message=message+" ";//Spacing
       }
       front=!front;
    }
    cout<<message.c_str();

    //If bottom is true
    if(printBm)//Bottom borders
    {
       cout<<"Ⅲ"<<endl;//Side Borders
       cout<<"⊙▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▪▪▪▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮⊙"<<endl;//Bottom borders
    }
    else
    {
       cout<<"Ⅲ"<<endl;//Side Borders
    }
}

//Making the Hang Man
void Hangman(float guesCnt=0)
{
    if (guesCnt>=1)//First Fail
       PrntMsg("|", false,false);//First part of the rope
    else
       PrntMsg("",false,false);//Spacing

    if (guesCnt>=2)//Second Fail
       PrntMsg("|",false,false);//Second part of the rope
    else
```

```
        PrntMsg("",false,false);//Spacing

    if (guesCnt>=3)//Third Fail
        PrntMsg("Q",false,false);//Head of Hang Man
    else
        PrntMsg("",false,false);//Spacing

    if (guesCnt==4)//Fourth Fail
        PrntMsg("/  ",false,false);//Left arm of Hang Man

    if (guesCnt==5)//Fifth Fail
        PrntMsg("/| ",false,false);//Left arm and body of Hang Man

    if (guesCnt>=6)//Sixth Fail
        PrntMsg("/|\\",false,false);//Left arm, right arm, and body of Hang Man
    else
        PrntMsg("",false,false);//Spacing

    if (guesCnt>=7)//Ninth Fail
        PrntMsg("|",false,false);//Second part of the body
    else
        PrntMsg("",false,false);//Spacing

    if (guesCnt==8)//Eighth Fail
        PrntMsg("/ ",false,false);//Left leg of Hang Man

    if (guesCnt>=9)//Ninth Fail
        PrntMsg("/ \\",false,false);//Left and right leg of Hang Man
    else
        PrntMsg("",false,false);//Spacing
}

//Printed Letters
void PrntLtr(string input,char from,char to)
{
    string s;
    for(char i=from;i<=to;i++)//Letter inputs
    {
        if(input.find(i)==string::npos)//Found "npos" in chapter 19
```

```
        {
            s+=i;
            s+=" ";
        }
        else
            s+="  ";
    }
    PrntMsg(s,false,false);
}

//Available Letters to choose from
void AvbLtrs(string taken)//Available Letters to type
{
    PrntMsg("Available letters");//Shows Letters to choose from
    PrntLtr(taken,'A','G');//Letters A - G
    PrntLtr(taken,'H','M');//Letters H - M
    PrntLtr(taken,'N','T');//Letters N - T
    PrntLtr(taken,'U','Z');//Letters U - Z
}

//Checking if the user won
bool WinChck(string word, string guessed)//Checks if user won
{
    bool won=true;//Set won to true
    string s;
    for (int i=0;i<word.length();i++)//Word length
    {
        if (guessed.find(word[i])==string::npos)//finds word
        {
            won=false;//won set to false
            s+="_ ";
        }
        else
        {
            s+=word[i];//If += to word
            s+=" ";
        }
    }
    PrntMsg(s,false);//Prints message if false
```

```
    return won;//Returns win
}

//Loading the random word
string LoadWrd(string path)
{
    int lineCount=0;//Set line count to 0
    string word;//String for word
    vector<string> v;//found the use of vectors in Chapter 8,
                     //v is used for word line
    ifstream reader(path.c_str());
    if(reader.is_open())
    {
        while(std::getline(reader,word))
            v.push_back(word);
        int rndmLne=rand()%v.size();
        word=v.at(rndmLne);
        reader.close();
    }
    return word;//Returns word
}

//Tries left in game
int TrysLft(string word,string guessed)
{
    int error=0;//setting error to 0
    for (int i=0;i<guessed.length();i++)//Having 'i' equal to zero while having it
                                        //less than the 'guessed.length'
    {
        if (word.find(guessed[i])==string::npos)
            error++;
    }
    return error;//Returns the error
}

//User Libraries

//Global Constants - Math/Physics Constants, Conversions,
//                   2-D Array Dimensions
```

```
//Function Prototypes

//Execution Begins Here
int main(int argc, char** argv) {

   //Players info
   string name; //Players name
   int age;     //Players age

   cout<<"Enter your name: ";//Outputs players name
   cin>>name;                //For player to type in name
   cout<<"Enter your Age: "; //Outputs players age
   cin>>age;                 //For player to type in age

   Player user(name, age);   //users name

   cout<<endl<<"Players Name: "<<user.getName()<<endl<<
           "Age: "<<user.getAge()<<endl;

   cout<<endl;

   cout<<"Enter Player 2's name (If no player 2, type 'n/a'): ";//Outputs player 2's name
   cin>>name;                                          //For player 2 to type in name
   cout<<"Enter player 2's Age (Type 0 if none): ";           //Outputs player 2's age
   cin>>age;                                          //For player to type in age

   Player user2;       //New player code

   user2.setName(name);//Player 2's name
   user2.setAge(age);  //Player 2's age

   cout<<endl<<"Player 2's Name: "<<user2.getName()<<endl<<
           "Age: "<<user2.getAge()<<endl;

   cout<<endl;

   cout<<"Hello "<<user.getName()<<" (Age: "<< user.getAge()
             <<") and "<<user2.getName()<<" (age: "
```

```
            << user2.getAge()<<")!"<<endl;

//Random number generator
srand(time(0));

//Declare Variables
string guesses;//Guesses
string wrdGues;//Word Guesses

//Process/Map inputs to outputs
   wrdGues=LoadWrd("Words.txt");//Word file
   int tries=0;               //Tries
   bool win=false;            //Win equal to false
   do                        //do loop
{
      system("clear"); //Fresh system
      PrntMsg("Welcome to Hang Man");//Intro to Hangman
      Hangman(tries);              //Amount of tries player has attempted
      AvbLtrs(guesses);            //Available guesses left
      PrntMsg("Guess the Word (Caps only)");    //Tells player to guess to word
      WinChck(wrdGues,guesses);     //Checks whether you won or not
      if(win)
         win=1;
      char x;
      cout<<">";cin>>x;            //User input
      if(guesses.find(x)==string::npos)
        guesses+=x;
      tries=TrysLft(wrdGues,guesses);//?gives you the amount of tries left
   }while(tries<10);
   if(win)
      PrntMsg("You Win");//Text if you won
   else
      PrntMsg("You Lost. Correct Word Was:");//Text if you lost
      cout<<setw(22)<<wrdGues<<endl;
      float border=1,border2=2;//Border 1 and border
      switch(1){
         case 1:cout<<"⊙▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▣"; //Extra borders
         case 2:cout<<"▣▣▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮⊙";//Extra borders
      }cout<<border<<border2;
```

```
    //Exit stage right!
    return 0;
}
```

Player.cpp:

```
/*
 * File:   main.cpp
 * Author: Blake Ward
 * Created on April 17, 2018, 7:30 PM
 * Purpose:  A game of Hang Man.
 */
//Function Definitions

#include "Player.h"

Player::Player(){
    newAge = 0;
}

Player::Player(string name, int age){
    newName=name;
    newAge=age;
}

Player::~Player(){

}

string Player::getName() const{
    return newName;
}

int Player::getAge() const{
    return newAge;
}
```

```
void Player::setName(string name){
   newName=name;
}

void Player::setAge(int age){
   newAge=age;
}
```

Player.h:

```
//Header ==> Function Declarations
/*
 * File:   main.cpp
 * Author: Blake Ward
 * Created on April 17, 2018, 7:30 PM
 * Purpose:  A game of Hang Man.
 */

#include <iostream>
#include <string>

using namespace std;

#ifndef PLAYER_H
#define PLAYER_H

class Player {
public:
   //Default Constructor
   Player();

   //Overload Constructor
   Player(string, int);

   //Destructor
   ~Player();

   //Accessor Functions
```

```
    string getName() const;
          //getName - returns name of player

    int getAge() const;
       //getAge - returns age of player

    //Mutator Functions
    void setName(string);
        //setName - sets name of player
        //@param string - name of patient

    void setAge(int);
       // setAge - sets age of player
       // @param int - age of player

private:
    //Member Variables
    string newName;
    int    newAge;
};

#endif /* PLAYER_H */
```