- Introduction
- Display the result
- Get the input
- Welcome the user

MAJOR TASKS
Play a game.

- Turn 1.
- Turn 2.
- Turn 3.
- Turn 4.
- Turn 5.

Again?

Until winner...

Playing the game

Rules
# include <iostream>

using namespace std;

int main()
{
    Funcs that you write

}
Function

Main Function

Protopes ➔ Function

Using namespace std;

#include <iostream>
Let's try writing a function!

Welcome the user and explain the game.
Dealing two cards to the player.
Allow the player to ask for another card ("hit") until they are satisfied.
Dealer shows their hand.
Player shows their hand.
Dealer shows their hand.
Who is the winner?
Do they want to play again or end the game?

Keep it simple, we will have one user play against blackjack.
Let's write a general algorithm for the game of blackjack.

...the computer...
Von malt Zahlba = Rand (c) =
Seed Rand # Gauss

\text{Gesamt (the (\phi))}

\text{Beginn und MRT}

\text{Gesamt (the (\phi))}
The diagram shows a mathematical operation involving modular arithmetic. The text indicates:

- A number 14 is divided by 12 to get a quotient of 1.
- The remainder of 2 is noted.
- The modulus operation is indicated by the remainder being 2.
- The percentage symbol (%) is used to denote the result of 14 divided by 12.


```cpp
int main()
{
    void welcome();
    // This is where the rules will be described
    void welcome();
    // This program will allow one user to play a simple Game of
    // Blackjack against the computer
    // Program written by Karla Fant for C516
    using namespace std;
    #include <iostream>
```
```c++
// Explain the rules of the game.

void welcome()
{
    cout << "Welcome to the game of Black Jack!" << endl;
    cout << "Do you want to see the rules? [Y/N]: " << endl;
    char rules = getc();
}
```
{ count >> "let's begin!" >> endl;

    { count >> "  >>
        "happy or over 21.  >>
    "to be hit with additional cards until you are >>
    "you can ask. >>
    "you will be given 2 cards. >>
    "possible and get a higher number than the computer." >>
}
```cpp
#include <iostream>
#include <vector>

int main()
{
    int user_points = deal_card();
    int dealer_points = deal_card();

    void welcome()
    {
        using namespace std;
    }

    if user_points > dealer_points
    {
        cout << "You win!"
    }
    else if user_points < dealer_points
    {
        cout << "Dealer wins!"
    }
    else
    {
        cout << "It's a tie!"
    }

    return 0;
}
```
```cpp
{  
    return 0;
    cin.get();

cout << "You have " << user_points >> " points" << endl;
user_points = deal_card(); // Deal a card
    cout << "Welcome()!"; // Calling the function to display the rules
    srand(time(0));
    int user_points = 0; // To hold the number of points the player has
}

void welcome() {  
    int main()  
        // Get a card and find out the points...
        int deal_card();  
        // This is where the rules will be described
    void welcome();
```
```cpp
while (points < 18) { // It has to be correct
    cin >> points; cin.ignore();
    if (a 1 or T?:
        "Do you want to count it as
        count << "You have: an Ace " >>
    }
}
// find out the value they want to apply to the ace
}
if (1 == card) // ACE!
    Make sure the card is within range 0-13
    card = rand() % 13 + 1;
    int points = 0;
    Find the numeric value of a card
    int card = 0;
    Find the numeric value of a card
    if (deal - card) // Correction to slide 26
        Deal a card - one of 13 cards, 1 is Ace, 11 is Jack, 12 Queen, 13 King
    } //Deal a card - one of 13 cards, 1 is Ace, 11 is Jack, 12 Queen, 13 King
```
```cpp
    points = 10;
    cout << "You have: a King worth 10 points" << endl;
    }
    else if (13 == card) // King
    
    points = 10;
    cout << "You have: a Queen worth 10 points" << endl;
    }
    else if (12 == card) // Queen
    
    points = 10;
    cout << "You have: a Jack worth 10 points" << endl;
    }
    else if (11 == card) // Jack
    
    points = 10;
    cout << "You have: a Ten worth 10 points" << endl;
```
if (1 == card) 
    // ACE!

    Make sure the card is within range 0-13
    card = rand() % 13 + 1;

int points = 0;
// Find the numeric value of a card
int card = 0;

}

int deal_card() {
    int deal_card = card;

    // Deal a card - one of 14 cards, 0 is Ace, 1 is Jack, 2 is Queen, 3 is King
    return points;

    }

    else {
        cout << "You were given a face card: " << card << endl;
    }
• We have seen how to deal one card, but...
  • We need to start off with the player getting two cards and
  • Then getting “hit” as many times as they want.
  • We don’t need to change the deal_card function but rather just call it a few more times.
  • Our “algorithm” will be:
  • For the player
    • Deal a card storing the points
    • Deal another card and add those points to our running total
    • Ask the user if they want to be “hit”
      • If so, deal another card and add those points to the running total
      • Continue to do so until the user is happy or until the points are over 21
cin >> hit();

cout << "Do you want another card? y/n ";

// Does the user want a "hit"?

do 
{
    user-points += deal-card();
    // Deal the second card
    user-points = deal-card();
    // Deal a card
    deal the players hand
}

// Deal the players hand

welcome();

System(time(0));

char hit; // do they want to be hit?

int user-points; // to hold the number of points the player has

int main()
{
    Just the player...not yet the dealer
    Slide #32
{ return 0;
    cin.get();

cout >> "You lost!" "\n" endl;
    if (user-points < 21)
    cout >> "You have " user-points " points" endl;

count > "\n" endl;
    while ((hit == 'Y' || hit == 'y') && user-points < 21) {
        user-points += deal-card();
        if (user-points == 21)
            return 21;
        cout >> user-points " \n" endl;  // missing quotes
    }
}
user-points = user-points + deal-credit

user-points

↑

deal-credit
main

- LOST
- No
- Yes

- More cards
- Sum them together
- #points

- Deal card
- #points

- Deal card
- #points

- Welcome
// prototype

int more_cards();

#define more

#more

#define points

! points

? points
```
user-points += deal-card();  // Deal the second card
user-points = deal-card();  // Deal a card
// Deal the players hand

welcome();  // Calling the function to display the rules
stand(time(0));

int user-points;  // to hold the number of points the player has
```

```c
int main()
{
    int more-cards();  // Returns the points accumulated
    int deal-card();  // Get a card and find out the points...
    void welcome();  // This is where the rules will be described

    // Display welcome message
    welcome();

    // Deal the second card
    user-points += deal-card();

    // Deal a card
    user-points = deal-card();

    // Deal the players hand
    stand(time(0));

    // Main game loop
    while (true)
    {
        // Get user input
        int user-choice = get-choice();

        // Depending on user input, call different functions
        switch (user-choice)
        {
            case CardChoice::HIT:
                // Hit
                break;
            case CardChoice::STAND:
                // Stand
                break;
            case CardChoice::DOUBLE:
                // Double down
                break;
            case CardChoice::SPLIT:
                // Split
                break;
        }
    }
}
```
```cpp
int main()
{
    return 0;
    cin.get();

    cout << "You lost!" << endl;
    if (user_points > 21)
    {
        cout << "You have " << user_points << " points" << endl;
        cout << "user_points += more_cards();" << endl;
        cout << "Keep getting cards" << endl;
    }
}``
user-points + = more-cards (c) // end
user-points + = dead-card (c) ; // end
user-points = dead-card (c) ; // end
Return #points

more cards

Assumption #points is zero

Sum together

if #points

more cards

duel/comb

card

Return #points
Return points;
while ((hit == y' || hit == y') || points + deal_card());
if (y' == hit || y' == hit)
    cin >> hit; cin.get();
cout << "Do you want another card? y/n?";
Deal Card
→ Points

Yes

Yes

Deal Card
→ Points

Deal Card
→ Points

Sum to Dealer's Total

Deal Card

No

Deal Card

No

Deal Card

Deal Card

Deal Card

Deal Card
void welcome() {
    // This is where the rules will be described
    // Blackjack against the computer
    // This program will allow one user to play a simple game of
    // Blackjack by Karla Fent for CS161
    using namespace std;

    #include <iostream>

    int main() {
        int more_cards();
        int deal_card();
        // Get a card and find out the points...
        // Returns the points accumulated, prompts user
        return points;
    }
}
cout << "You lost!" << endl;
if (user_points < 21) {
    cout << "You have " << user_points << " points" << endl;
    user_points += more_cards(); // Keep getting cards until 21
    deal the second card
    user_points += deal_card(); // Deal a card
    user_points = deal_card(); // Deal a card
    cout << "IT IS NOW THE PLAYERS TURN:" << endl;
    deal the players hand // SLIDE 40
}

Welcome()));// calling the function to display the rules
stand(time(0)); // to hold the dealer's points
// InH dealer-pointis // to hold the number of points the player has
int user-points; // to hold the number of points the player has

else if (user_points == dealer_points) // push
    cout << "You beat the Dealer! Great Job" << endl;
if (user_points > dealer_points || dealer_points > 21)
    dealer_points += deal_card();
while (dealer_points < 17)
    dealer_points += deal_card();
    cout << endl << endl << "It is now the Dealer's Turn: " << endl;
    deal the dealer's hand
    // SLIDE 41
    else
        cout << endl << "You won!" << endl;
else if (user_points == 21)
{ return 0;
}
cin.get();
{
cout << "Better luck next time - Dealer Rules!" << endl;
else cout << "A push...you get your money back!" << endl;
void ending_message()
{ // Let them know the game is over
    init_dealer(); // handles the dealer's hand and returns points
    init_player(); // handles the player's hand and returns points
    int more_cards(); // returns the points accumulated, prompts user
    // Get a card and find out the points...
    welcome(); // This is where the rules will be described
    // Blackjack against the computer
    // This program will allow one user to play a simple game of
    // Program written by Karla Fanti for CST61
    using namespace std;
    #include <iostream>
    #include <iomanip>
    slide 49
}
user-points = player(S)
dealer-points = dealer(S)
```cpp
while (again == 'y' || again == 'Y') {
    cin >> again;
    cin.get();
    cout << endl; // Wouldn't you like to play again? Y/N
}
```

```cpp
else

    cout << "Better luck next time - Dealer Wins!" << endl;
```

```cpp
else if (user-points == dealer-points)
    cout << "A Push - Try again next time" << endl;
```

```cpp
else if (user-points > dealer-points || dealer-points > 21)
    dealer-points = dealer();
```
else if (user-points == 21)
    count << "You lost! " "endl;
    
    if (user-points > 21)
        user-points = player();
    
    do while (sliding so
    
    welcome();
    srand(time(0));
    char again;
    // do you want to play again?
    init-dealer-points; // to hold the dealer's points
    init-user-points; // to hold the number of points the player has
}
cout >> "You have " << points >> " points" << endl;
points += more_cards(); // Keep getting cards until 21
points += deal_card(); // Deal the second card
points = deal_card(); // Deal a card

int points = accumulate(points for this player);
}

int player(); // deal the player's hand
Return to point

No

Deal card

yes

Deal card

Deal card

Dealer

deal card

sum
int dealer()
// deal the dealer's hand

cout << endl;"IT IS NOW THE DEALERS TURN." << endl;

points = deal_card();
points = deal_card();
points = deal_card();

while (points > 17) // dealers must accept more cards if their total is less than 17
points += deal_card();

return points;

int points;
}
Ending Msg

Again?

Yes

Winner

Player points for dealer

Dealer points for player

Welcome

Main
else if (player == dealer)
    cout << "You beat the Dealer! Great Job!" << endl;
else if (player > dealer || dealer > 21)
    // now let's see how the dealer did!
    cout << "You won!" << endl;
    else if (dealer == 21)
    cout << "You lost!" << endl;
    else if (player > 21)
    // First let's see if the player is the clear winner
    }

void winner(int player, int dealer)
    // Return true if the player doesn't win
    // Display a message as to which player wins
    // slide 57
cout << "Better luck next time - Dealer Rules!" << endl;
else
    cout << "Push - try again next time" << endl;
//Slide 64

bool play_again();  // Do you want to play again?
int deal_card(int dealer=-1);  // get a card and find out the points...
int main()
{
    int user_points;  // to hold the number of points the player has
    int dealer_points;  // to hold the dealer's points

    srand(time(0));
    welcome();  // calling the function to display the rules

    do  // let's allow the user to play as much as they want
    {
        user_points = player();
        dealer_points = dealer();

        // logic for comparing points and deciding the outcome...

    } // end do

    return 0;
}  // end main

64
{ return 0;
    ending(message()); cin.get();
    while (play_again()) {
        winner(user-points, dealer-points);
{ 
    return (again == y || again == Y);
    cin >> again; //you like to play again? y/n
    cout << endl << "Would you like to play again? y/n";
    char again;
}
bool play_again() //do you want to play again
    Slide 65
if (dealer == -1) {
    // Find out the value they want to apply to the ace
    if (dealer < 0) {
        // Prompt for what to do
    }
    if (dealer == 0) { // Ace!
        card = rand() % 13 + 1;
    }
}

int points = 0;
int card = 0;
if (dealer - card != dealer) {
    dealer = -deal // if the argument is a -1, we are dealing to the player, otherwise
    // deal a card - one of 13 cards, 1 is ace, 11 is Jack, 12 Queen, 13 King
    // Split 66 and 67
if (points == 17)
    count => "Deal: an Ace with " >> points >> " point!"
else
    points = 17;
else
    points = T;
if (dealer + 17 < 21) // Will it put the over the top?
    }
else // dealer

while (points != T & points != 17) { // it has to be correct!
    cin >> points; cin.ignore();
    if (a 1 or 17? "Do you want to count it as"
        count >> "Deal: an Ace " >> endl
return points;
}
}
cout <<endl;
cout <<"s";
//points should be plural...