```cpp
#ifndef _PLAYER_H_
#define _PLAYER_H_

#include <string>
#include <time.h>
#include <iostream>
using namespace std;

class Card;
class RatATat;

class Player
{
protected:
    string m_player_name;                                   //player name
    RatATat* m_player_pointer_to_ratatat;
    Card** m_player_card_arr;                               //player hand (array that holding cards pointers! ,size 4)

    //Extras:
    unsigned int m_player_cards_sum;                        //player cards sum

    static const int m_player_min_card_indx = 0;
    static const int m_player_max_card_indx = 3;
    unsigned int m_indx_to_put_a_card_in_player_hand;       //for method put_card_in_hand . m_indx_to_put_a_card_in_player_hand is first 0 , then after putting a card in player hand it up by 1

public:
    Player(string name , RatATat* pointer_to_game_ratatat); //ctor - initialize player name and player_cards_sum to 0 , //?and initialize every pointer to point to NULL ??//
    ~Player();          //dtor - delete : player_pointer_to_ratatat , and pointer player_card_arr (every card pointer in it will be deleted in pile (like the bj game)!!!
    //using deful cctor //??//
    friend ostream& operator<<(ostream& os, const Player& player); // print name of the player and his hand (his cards)
    //Virtual methods
    virtual int ChooseOption_from_card_menu(Card& currentCard) = 0; // virtual mehtod to choose an option from card_action_menu by AI_player and Human_player

    const static unsigned int m_player_number_of_cards = 4;          //The number of cards per player // to do a get!!!

    //getter and setters and "Extras":
    string get_player_name();                                    //return the player name
    unsigned int get_player_cards_sum() const;                         //return the cards sum of the player
    void put_card_to_player_hand(Card* card); //puts the card in player
```

```cpp
hand (for first ditrbution)
    void calc_sum_of_player_hand();
    void show_player_hand_and_make_it_valid_to_calc_sum();  //show player hand , and if there are special_cards , swap it with card from pile and show hand again
    Card* get_card_pointer_by_indx(int indx); //return card pointer by recieving indx of card in player_card_arr
    //void show_card_by_indx(int indx);
    void replace_cards(int indx_of_card_to_replace, Card*& card_b_pointer);//for using play_card , its ask indx to card of player to swap with
    //static , friends and virtual members and methodes:
    static int get_player_min_card_indx(); //0
    static int get_player_max_card_indx(); //3
    void print_msg_choose_card_indx() const;
    //virutal methods:
    virtual int choose_card_indx() = 0; //virtual method for choosing card_indx 0-3 (if AI -random choosing , if Human ask the player)
    virtual int choose_player(Player** players_arr , int curr_player) = 0;//method for choosing player for swap card with.
    virtual int player_choice_from_what_pile_to_pick( int number_of_option) = 0; //return 1 if player want to pick from unused pile , and 2 if he want to pick from discarded pile
    virtual int player_answer_to_call_ratatat( int number_of_option) = 0; //1=No - so return 0 . 2=Yes - so return 1;
};

class AI_player : public Player
{
public:
    AI_player(const string& player_name, RatATat* pointer_to_game_ratatat ) : Player(player_name, pointer_to_game_ratatat) {};
    int ChooseOption_from_card_menu(Card& currentCard) ; //geting a card , than show to the player (in this case the dealer) the card's action menu (throw the card to the pile or use it), then return randomly his answer
    int choose_card_indx() ;
    int choose_player(Player** players_arr, int curr_player) ;
    int player_choice_from_what_pile_to_pick(int number_of_option) ;
    int player_answer_to_call_ratatat(int number_of_option);
};

class Human_player : public Player
{
public:
    Human_player(const string& player_name, RatATat* pointer_to_game_ratatat) : Player(player_name, pointer_to_game_ratatat) {}; //ctor
    int ChooseOption_from_card_menu(Card& currentCard) ; //geting a card , than show to the player the card's action menu (throw the card to the pile or use it) , than ask for the human to "answer" and return it
    int choose_card_indx() ;
    int choose_player(Player** players_arr, int curr_player)  ;
    int player_choice_from_what_pile_to_pick( int number_of_option) ;
    int player_answer_to_call_ratatat( int number_of_option) ;
};


#endif // !_PLAYER_H_
```