

Programming Design Worksheet - Redfield

for CS1310 (programs 2-7) and CS1311 (programs 1-6)

Copy this file. Type and past images to create new documents for designs.
Print it for class (if you must miss, submit one file to Designs).

First name **Davide** Last name **Russillo**

Design for program name **Database**

DATA

Variables needed in WORDS for main and globally

global:

structure of a videogame

 includes name, space, playtime, price, and pointer to next game

pointer to head of linked list

pointer to last of linked list

pointer to current selected element of linked list

number of elements in linked list

main:

string selecting if you want to continue or not (starts at "")

C DECLARATIONS for main & global

global:

```
struct videogame {
```

```
    char name[31];
```

```
    double space; // in GB
```

```
    int playtime; // in hours
```

```
    double price; // in USD
```

```
    struct videogame *next;
```

```
};
```

```
struct videogame *head, *last, *current;
```

```
int entries;
```

main:

```
char selection[5] = "";
```

(STARTING TicTacToe: put image; or draw: Insert, Drawing; or put at end of the file)

draw in RAM with possible values

Algorithm to PSEUDOCODE level for each function

(remember to indent under if, switch, while, do-while, for)

main:

```
clear page
print This program take videogames' names, required space, playtime, and
      price as input. It then puts them in a liked list database.
print Press enter to continue
catch newline

while selection isn't done, Done, or DONE
    call print_nodes
    call add_node
    call print_nodes
    print would you like to continue? y for yes, done to quit.
    input into selection
```

other functions (bold the names): (put them before main in the program!)

```
void add_node(void)
    set valid_value to = 0 (false)
    point current to memory for new entry

    print enter name up to 30 characters
    get name from user and store in current name
    if no newline in current name
        clear input buffer with while loop
    else
        replace newline with null character

    while valid_value is zero
        print enter space in GB up to 9,999.99
        input float into current space
        if current space is 0 or less or greater than 9,999.99
            print invalid value was entered
        else
            set valid_value to 1
    reset valid_value to 0

    while valid_value is 0
        enter playtime in hours up to 99,999
        input playtime to current playtime
        if current playtime less than 0 or greater than 99,999
            print invalid value was entered
        else
            set valid_value to 1
    reset valid_value to 0
```

```

while valid_value is 0
    enter price in USD up to 999.99
    input price to current price
    if current price less than 0 or greater than 999.99
        print invalid value was entered
    else set valid_value to 1
reset valid_value to 0

clear buffer again to catch newlines

if head points to nothing
    set head to current
else
    set pointer to next of last to current
set last to current
increment entries by one

void print_nodes(void)
    clear page
    print CURRENT DATA (amount of entries)
    print
        | name | space | playtime | price |
    if head points to nothing
        print
            | NULL | NULL | NULL | NULL |
    else
        set current to head
        while current points to something
            print
                | name | space | playtime | price |
            set current to next

```