Indirect Access

Name: jasveena
Student ID: 224001588
Date: 1 -06 -2024

Learning Journey and Evidence

Review pointers, and references.

What did you think while reviewing this in the Programmer's Guide? Which aspects did you already know, which were challenging? Which explanations helped, which were confusing? Etc.

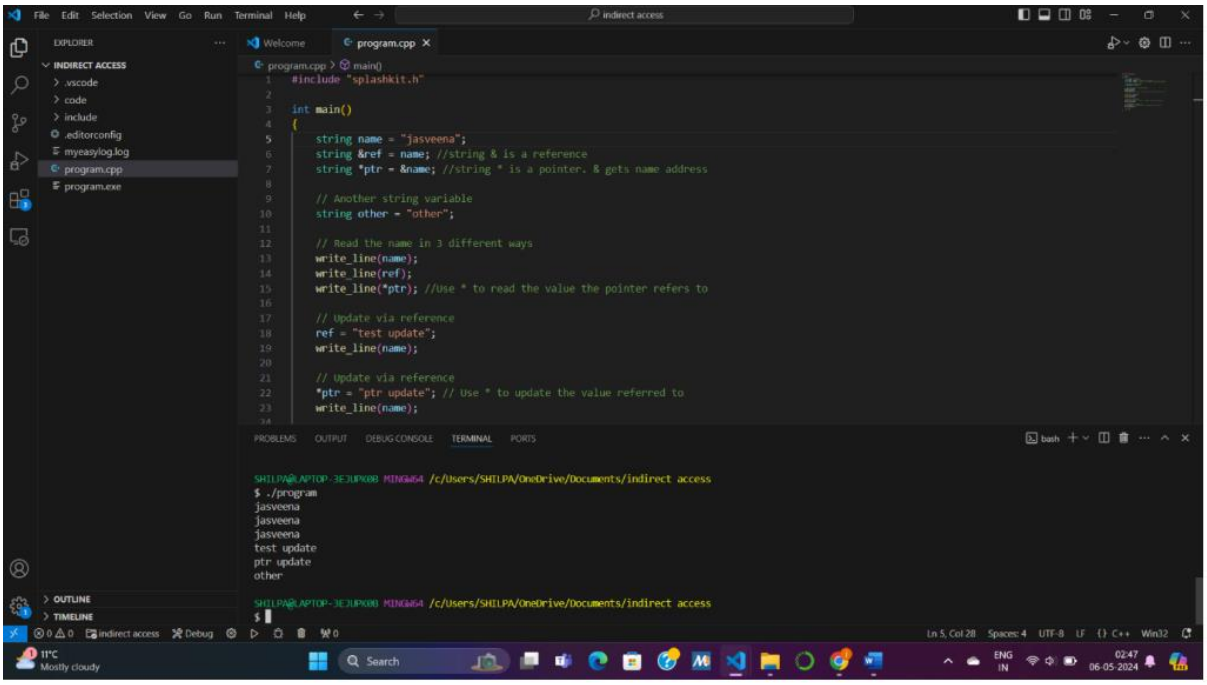
Pointers: A pointer is a data type built into the programming language. A pointer has a value, that stores the location of another value. The pointer's value is the memory address of the value it points to this means that we can point to any value in memory, regardless of where it is. A pointer value is the same as any other value it can be stored in local variables, global variables, it can be passed to a function in parameter and it can be returned from a function.
References:. Pointers and references both store the address of another value in memory. They are both used to refer to another value. While in pointer we need to manually get the address and store it in the pointer in reference the compiler takes care of this for us.

What other resources did you use to help understand these concepts? Any good resources you would recommend to others?

I used youtube to deepen my understanding of different headers

Create a small program to demonstrate the use of pointers.

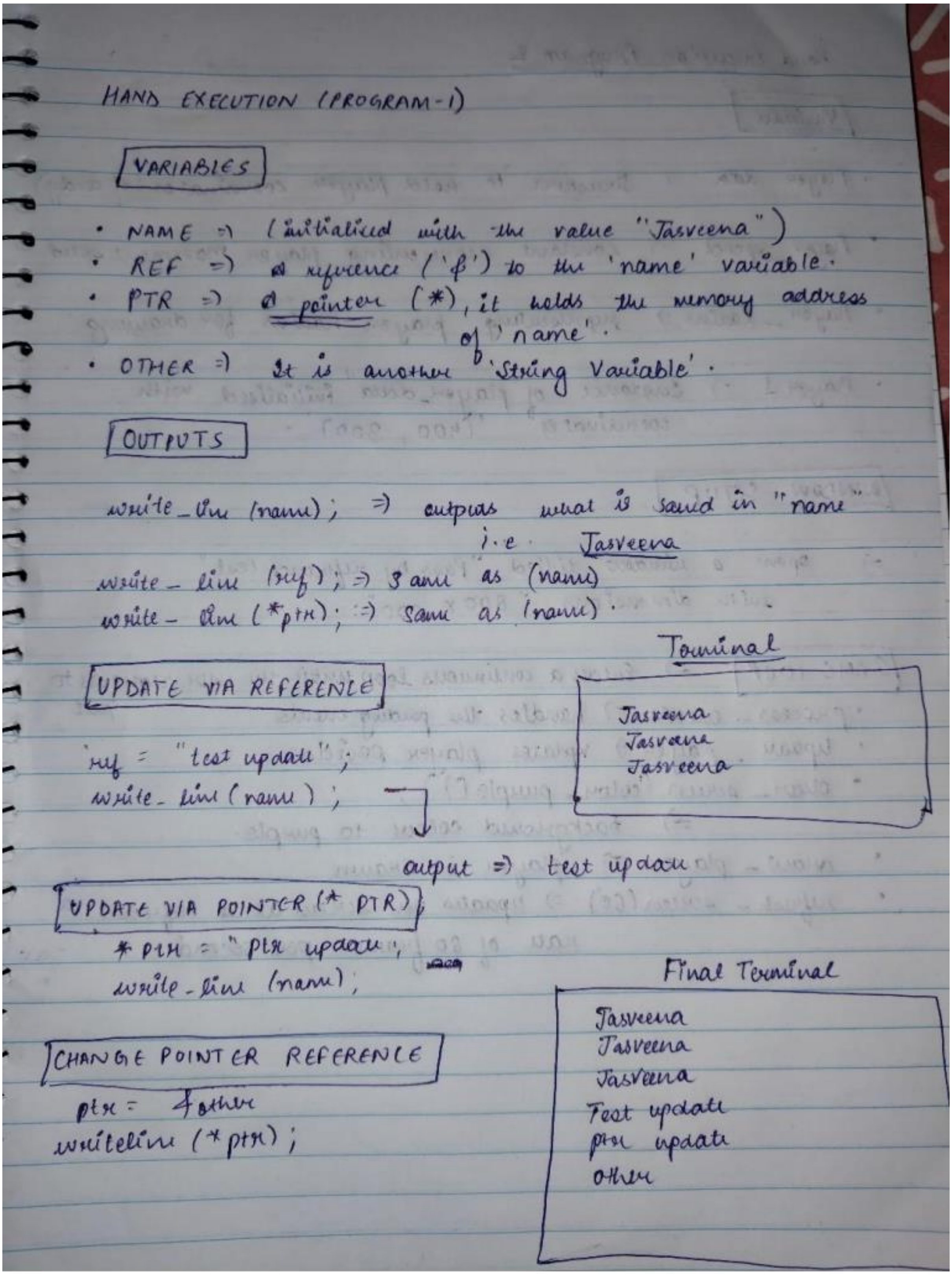
Show your process for this and highlight any realisations you gain.



```cpp
#include "splashkit.h"

int main()
{
    string name = "jasveena";
    string &ref = name; //string & is a reference
    string *ptr = &name; //string * is a pointer. & gets name address

    // Another string variable
    string other = "other";

    // Read the name in 3 different ways
    write_line(name);
    write_line(ref);
    write_line(*ptr); //Use * to read the value the pointer refers to

    // Update via reference
    ref = "test update";
    write_line(name);

    // Update via reference
    *ptr = "ptr update"; // Use * to update the value referred to
    write_line(name);
```

```
SHILPA@LAPTOP-3EJUPK0B MINGW64 /c/Users/SHILPA/OneDrive/Documents/indirect access
$ ./program
jasveena
jasveena
jasveena
test update
ptr update
other

SHILPA@LAPTOP-3EJUPK0B MINGW64 /c/Users/SHILPA/OneDrive/Documents/indirect access
$
```

Include a hand execution of the program to explore how it works.



HAND EXECUTION (PROGRAM-1)

VARIABLES

- NAME => (initialized with the value "Jasveena")
- REF => a reference ('&') to the 'name' variable.
- PTR => a pointer (*), it holds the memory address of 'name'.
- OTHER => It is another 'String Variable'.

OUTPUTS

write_line (name); => outputs what is saved in "name" i.e. Jasveena

write_line (ref); => same as (name)

write_line (*ptr); => same as (name).

Terminal

```
Jasveena
Jasveena
Jasveena
```

UPDATE VIA REFERENCE

```
ref = "test update";
write_line (name);
```

output => test update

UPDATE VIA POINTER (* PTR)

```
*ptr = "ptr update";
write_line (name);
```

CHANGE POINTER REFERENCE

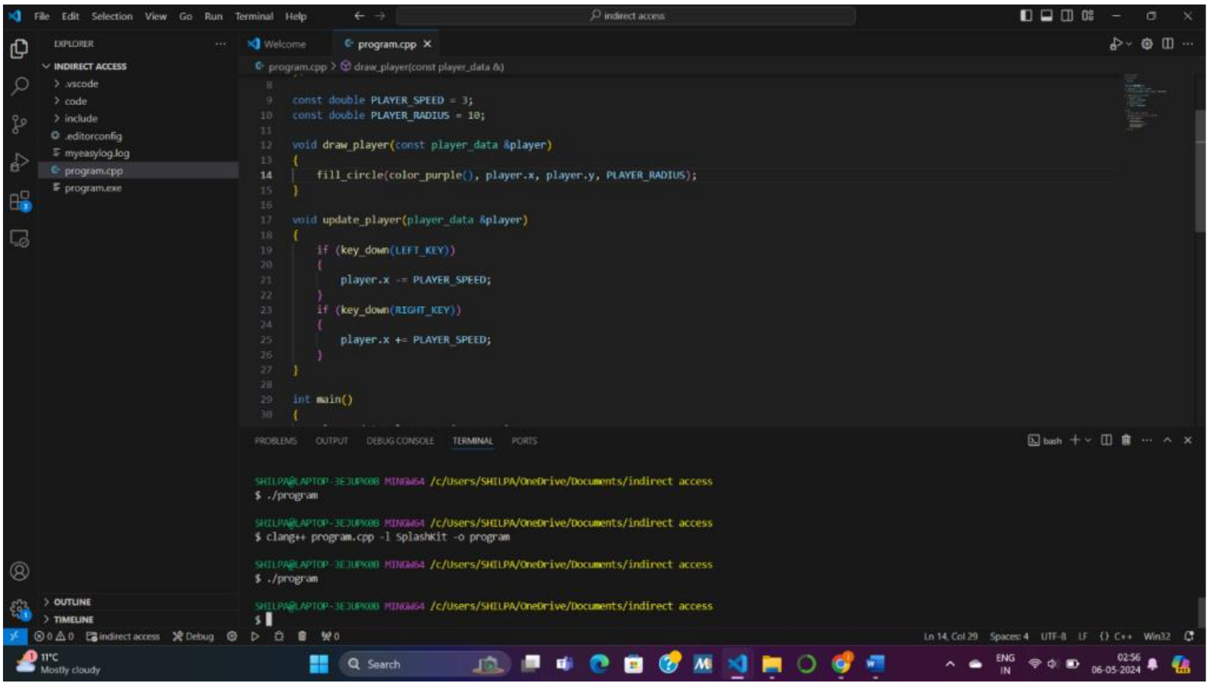
```
ptr = &other
writeline (*ptr);
```

Final Terminal

```
Jasveena
Jasveena
Jasveena
Test update
ptr update
other
```

Create a small program to demonstrate the use of pass-by reference.

Show your process for this and highlight any realisations you gain.



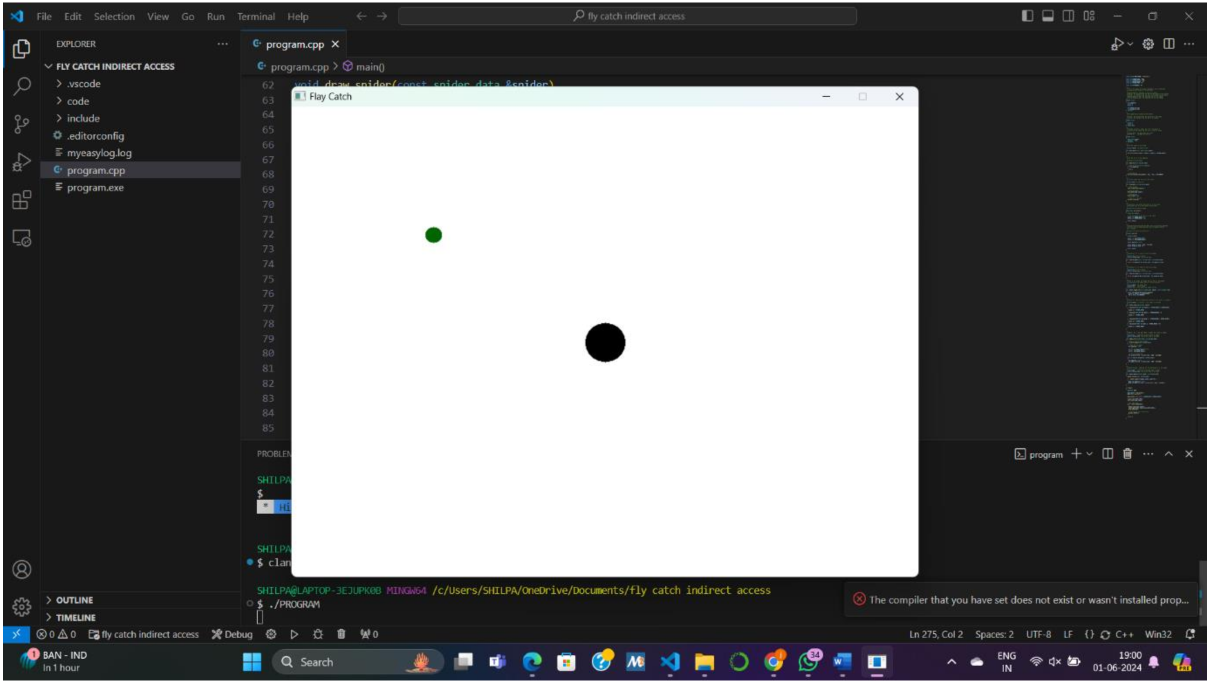
```
const double PLAYER_SPEED = 3;
const double PLAYER_RADIUS = 10;

void draw_player(const player_data &player)
{
    fill_circle(color_purple(), player.x, player.y, PLAYER_RADIUS);
}

void update_player(player_data &player)
{
    if (key_down(LEFT_KEY))
    {
        player.x -= PLAYER_SPEED;
    }
    if (key_down(RIGHT_KEY))
    {
        player.x += PLAYER_SPEED;
    }
}

int main()
{
```

```
SHILPA@LAPTOP-3EJUPK0B MINGW64 /c/Users/SHILPA/OneDrive/Documents/indirect access
$ ./program

SHILPA@LAPTOP-3EJUPK0B MINGW64 /c/Users/SHILPA/OneDrive/Documents/indirect access
$ clang++ program.cpp -l SplashKit -o program

SHILPA@LAPTOP-3EJUPK0B MINGW64 /c/Users/SHILPA/OneDrive/Documents/indirect access
$ ./program

SHILPA@LAPTOP-3EJUPK0B MINGW64 /c/Users/SHILPA/OneDrive/Documents/indirect access
$
```

Include a hand execution of one of the calls to swap to explore how it works.

Update your Fly Catch program to make use of pass by reference

Show your process for this and highlight any realisations you gain.



```
void draw_spider(const spider_data &spider)
```

```
SHILPA@LAPTOP-3EJUPK0B MINGW64 /c/Users/SHILPA/OneDrive/Documents/fly catch indirect access
$ ./PROGRAM
```

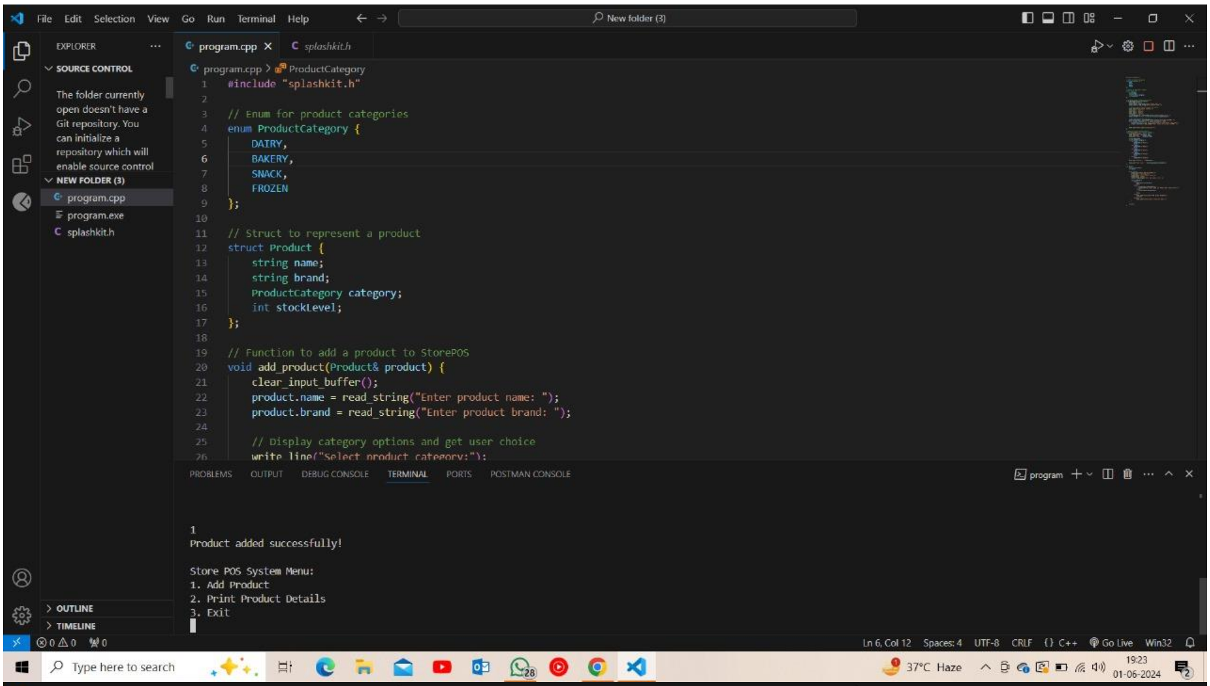
The compiler that you have set does not exist or wasn't installed prop...

Capture any other study and practice used to master these concepts.

Show evidence of any additional study and practice you did to master these concepts.

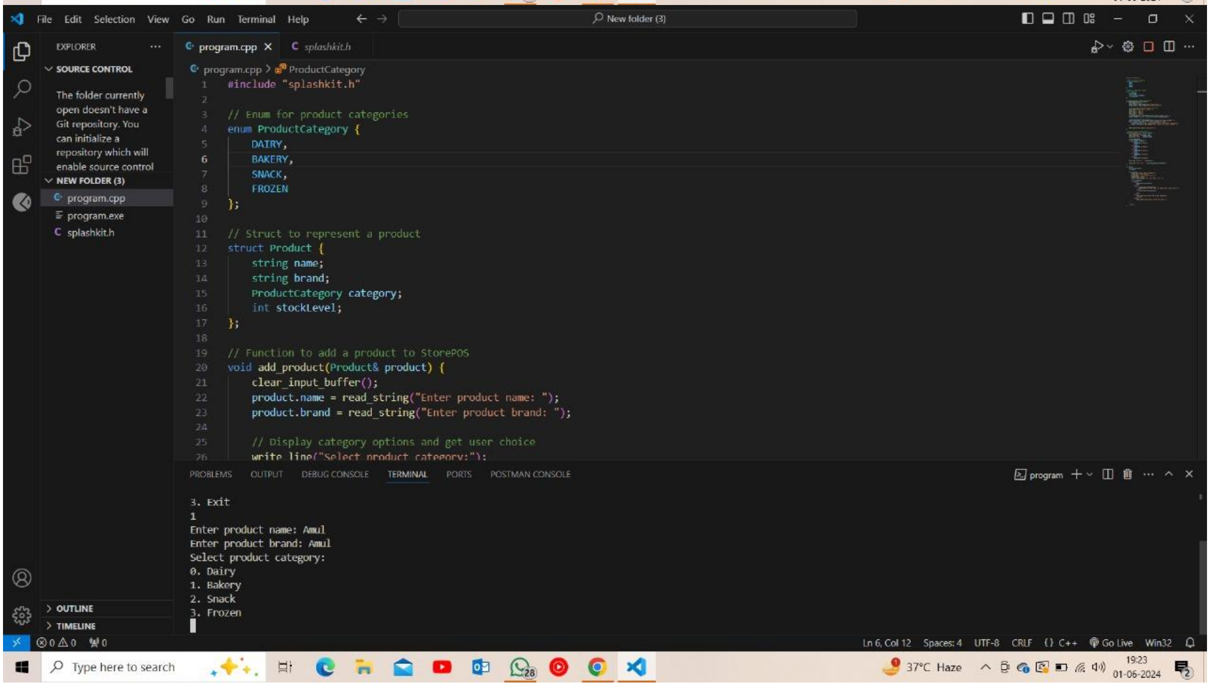
Complete one of the Test Your Knowledge activities

Show your process for this and highlight any realisations you gain.



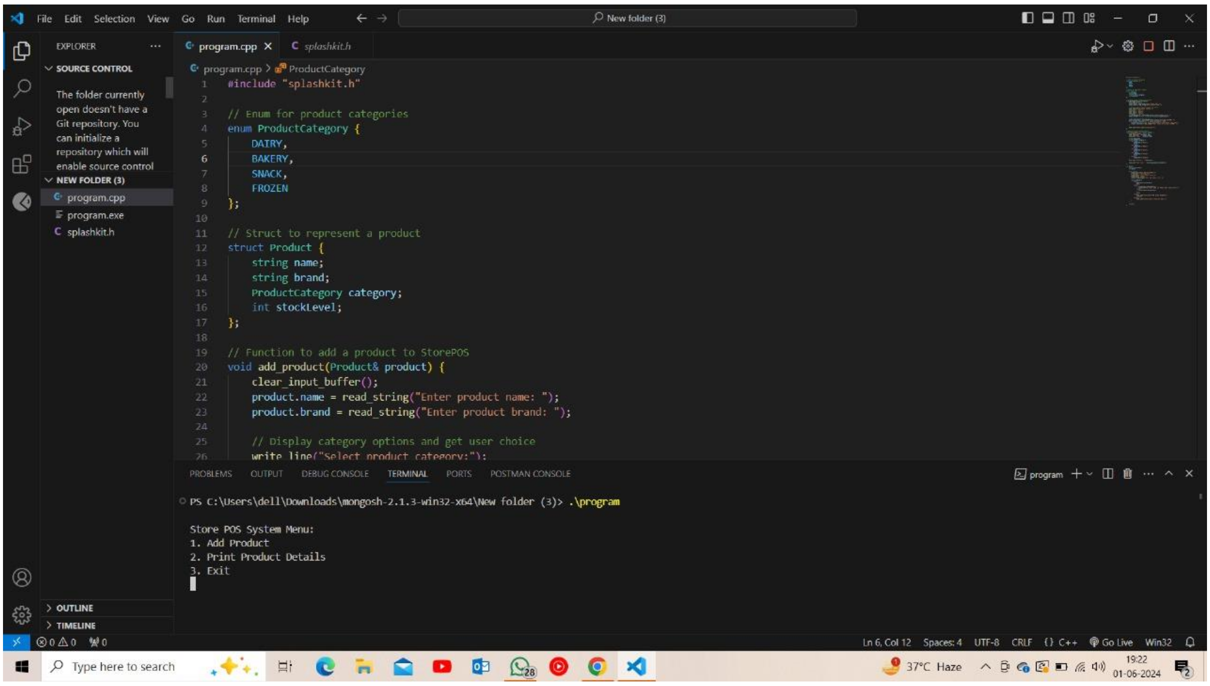
```cpp
#include "splashkit.h"

// Enum for product categories
enum ProductCategory {
    DAIRY,
    BAKERY,
    SNACK,
    FROZEN
};

// Struct to represent a product
struct Product {
    string name;
    string brand;
    ProductCategory category;
    int stockLevel;
};

// Function to add a product to StorePOS
void add_product(Product& product) {
    clear_input_buffer();
    product.name = read_string("Enter product name: ");
    product.brand = read_string("Enter product brand: ");

    // Display category options and get user choice
    write_line("Select product category:");
```

```
1
Product added successfully!

Store POS System Menu:
1. Add Product
2. Print Product Details
3. Exit
```



```cpp
#include "splashkit.h"

// Enum for product categories
enum ProductCategory {
    DAIRY,
    BAKERY,
    SNACK,
    FROZEN
};

// Struct to represent a product
struct Product {
    string name;
    string brand;
    ProductCategory category;
    int stockLevel;
};

// Function to add a product to StorePOS
void add_product(Product& product) {
    clear_input_buffer();
    product.name = read_string("Enter product name: ");
    product.brand = read_string("Enter product brand: ");

    // Display category options and get user choice
    write_line("Select product category:");
```

```
3. Exit
1
Enter product name: Amul
Enter product brand: Amul
Select product category:
0. Dairy
1. Bakery
2. Snack
3. Frozen
```


File Edit Selection View Go Run Terminal Help
New folder (3)
EXPLORER
SOURCE CONTROL
NEW FOLDER (3)
program.cpp
program.exe
splashkit.h
OUTLINE
TIMELINE
program.cpp
splashkit.h
program.cpp > ProductCategory
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
program
Ln 6, Col 12 Spaces: 4 UTF-8 CRLF {} C++ Go Live Win32
Type here to search
37°C Haze
19:22
01-06-2024

The folder currently open doesn't have a Git repository. You can initialize a repository which will enable source control

```cpp
#include "splashkit.h"

// Enum for product categories
enum ProductCategory {
    DAIRY,
    BAKERY,
    SNACK,
    FROZEN
};

// Struct to represent a product
struct Product {
    string name;
    string brand;
    ProductCategory category;
    int stockLevel;
};

// Function to add a product to StorePOS
void add_product(Product& product) {
    clear_input_buffer();
    product.name = read_string("Enter product name: ");
    product.brand = read_string("Enter product brand: ");

    // Display category options and get user choice
    write_line("Select product category:");
```

```
PS C:\Users\dell\Downloads\mongosh-2.1.3-win32-x64\New folder (3)> .\program

Store POS System Menu:
1. Add Product
2. Print Product Details
3. Exit
```

Brief Summary of Concepts

Concept	Key Idea / Concept
Pass by Value vs Pass by Reference	These are the terms that explain how data is passed to a parameter. Pass by value copies value to the parameter while pass by reference is like passing the variable itself to the parameter.
Pointers	It is a datatype built into the programming language
References	While in pointer we need to manually get the address and store it in the pointer in reference the compiler takes care of this for us.
Null	Indicates an invalid or non-binding value or associated with the value 0.
Segmentation Fault	if you attempt to dereference a pointer and perform an action that is not permitted at that location in memory, this will result in a segmentation fault.
Dangling Pointers	Dangling pointer is a pointer that does not point to na valid value.

Reflection

What gives you confidence you have achieved the learning goals?

Use pass-by reference to accept and update values

Passing parameters by reference allows you to modify the original variables passed to a function. I used the pass by reference to accept and update values in the program shown above.

*Use **const (constant)** references to provide read-only access to data to improve performance.*

Utilizing const references ensures that the data passed to a function cannot be modified within the function, which can improve performance by avoiding unnecessary copying of data. I Read through the programmers field guide and used it in the small program

Use pointers to refer to and interact with other values in memory.

Pointers are powerful tools for interacting with memory addresses and manipulating data indirectly. By creating programs that demonstrate the use of pointers, we'll gain a deeper understanding of how they work and their practical applications, such as dynamic memory allocation and data structures.

What is the most important thing you learned from this and why?

The most important thing i learned from this exercise is the fundamental concepts of pointers and references and how they differ in terms of syntax, behavior, and usage. Understanding these concepts is crucial for writing efficient and robust code, especially in scenarios where memory management and performance are critical.