Variables
And
Data Types

Vision of this week

We want to write a Program that takes Distance (in kilometers) travelled by a car in Time (hours) and calculates its Speed (kilometer/hour).

Vision of this week

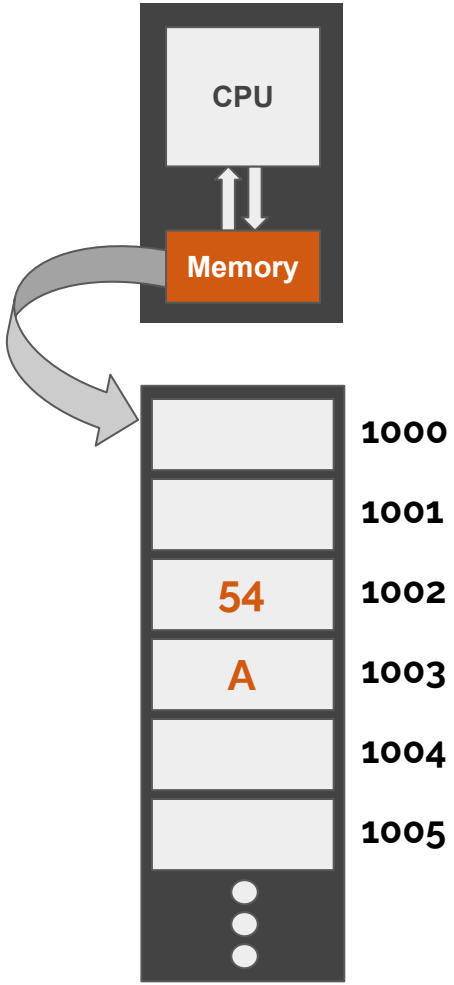
We want to write a Program that takes Distance (in kilometers) travelled by a car in Time (hours) and calculates its Speed (kilometer/hour).

C:\Windows\system32\cmd.exe

```
D:\>c++ second.cpp -o second.exe

D:\>second.exe
Enter distance..40
Enter time..10
Speed is 4
D:\>
```

Review: Main Memory

- Memory is called Main Memory, Primary Memory or RAM.
- This memory is divided into different cells.
- Each cell has an address like we have address of our house numbers or PO Boxes
- CPU stores data into these cells and loads data from these cells whenever it is required.



When do we need memory?

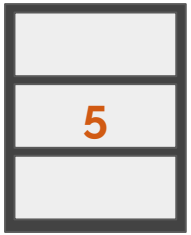
When do we need memory?

PURPOSE

When do we need memory?

1. When we take Input from any device, we need Memory to store the Data.

5

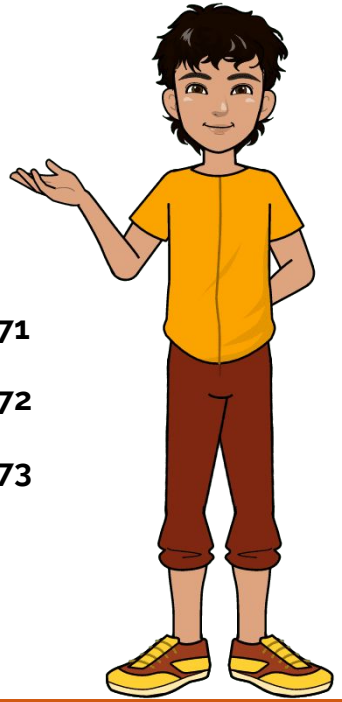


0xE4A71

0xE4A72

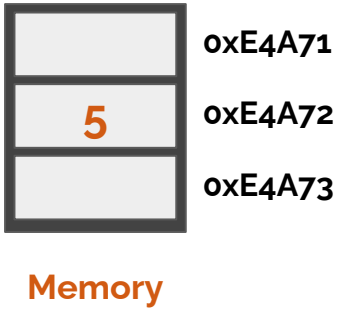
0xE4A73

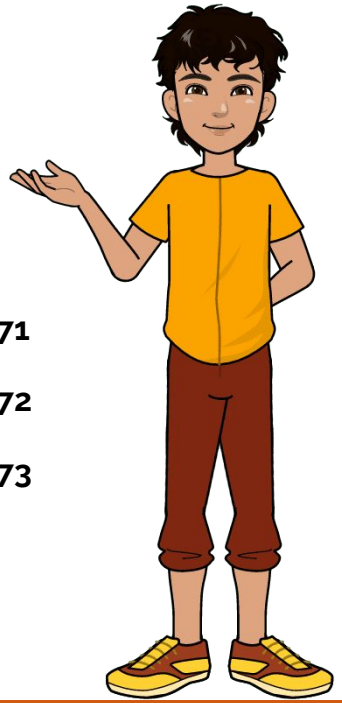
Memory



When do we need memory?

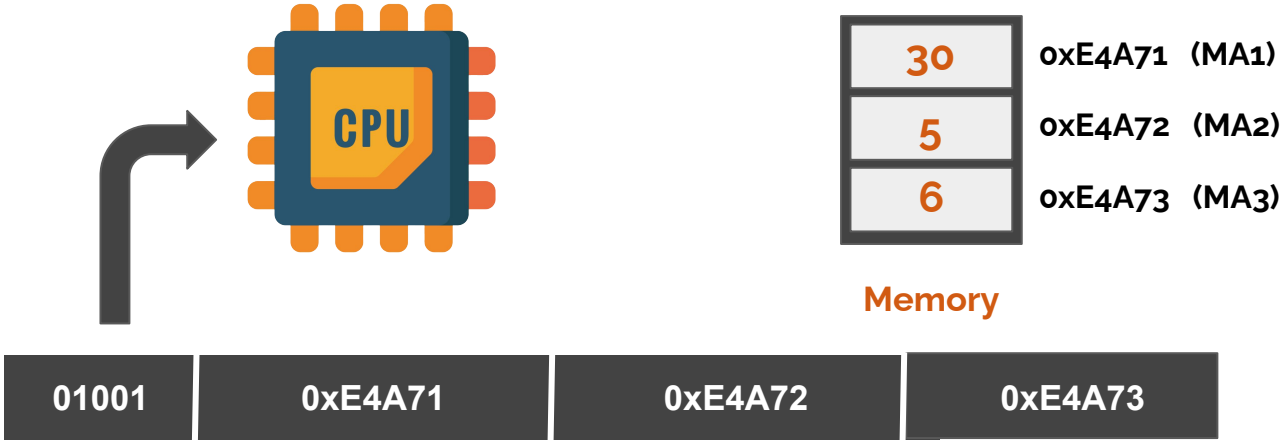
2. When we show Output on the screen, We need to load data from Memory



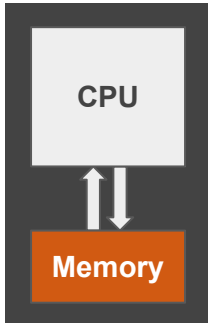


When we need memory?

3. CPU performs operations on the data that is in the Memory. Let 01001 is Operation Code for Division



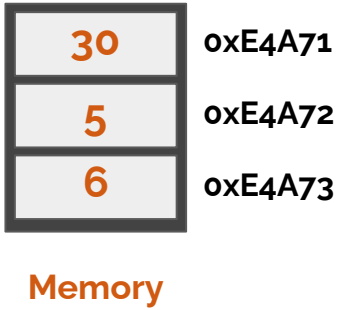
Review: Memory



- When CPU takes input from devices, it stores information into **memory** before processing it.
- CPU stores results of the processing into the **memory.**
- CPU stores information into the **memory** before sending it to output devices.

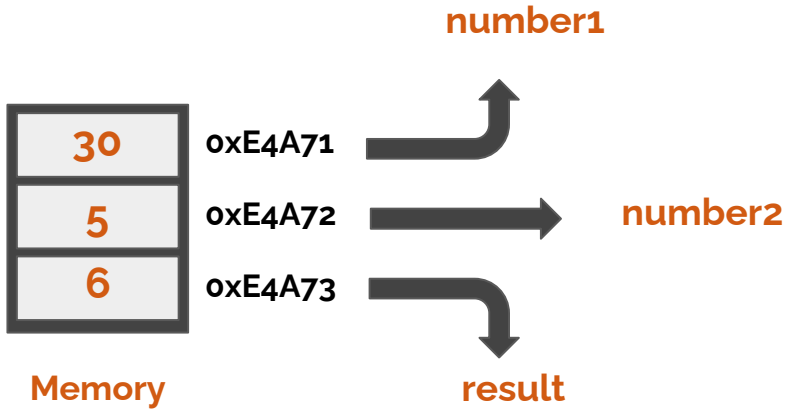
How to Allocate Memory: Variables

To store data into the Memory, we need to reserve the space in the Memory.
When the space is reserved, we can store or retrieve data from the Memory through its Memory Addresses.



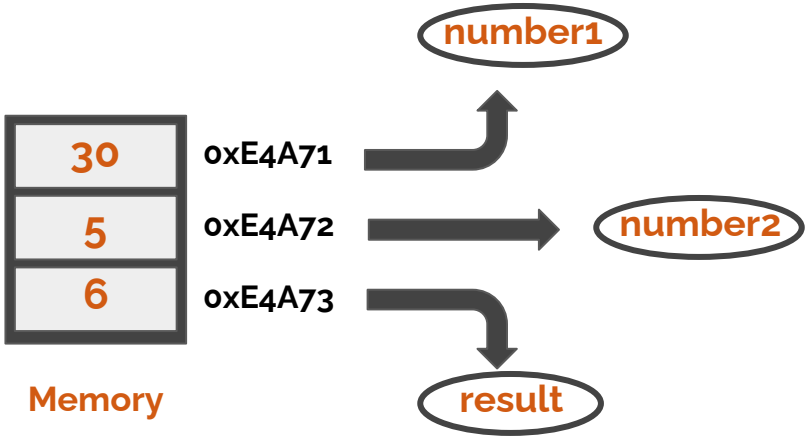
Variables: Names

It is difficult to remember the Addresses of these Memory locations.
High Level Languages allow us to give Names to these reserved Memory locations.



Variables: Names

These Names are called the Variables.
Variables are the names that we give to the Memory Locations.



Variables: Names

All High Level Languages apply some Naming Rules on the variables

- The name can not have Spaces
- The name can not start with Numbers
- The name can not have any Special Character (&, !, %, # etc)

Variables: Names

All High Level Languages apply some Naming Rules on the variables

These are some of the Valid names of the Variables

number1 num_1 num1

numb2 nu_2 _n2

result_1 Res _Res

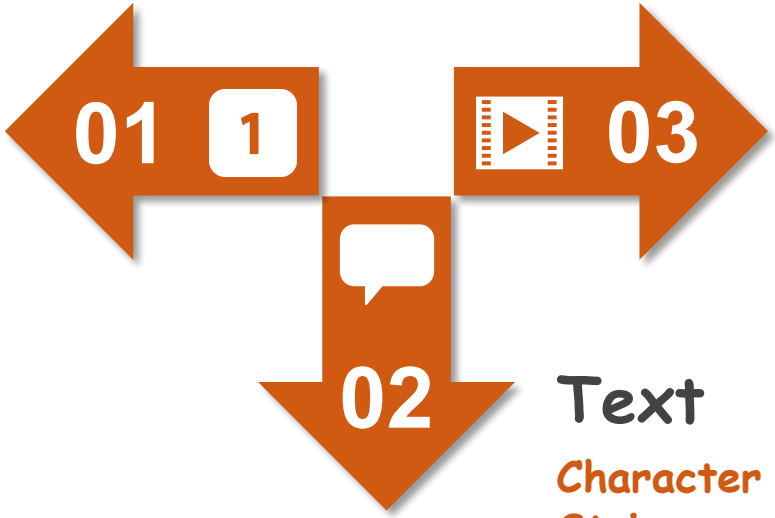
What Type of Data in Memory?

Now, We know that we can deal with memory using Variables. But the question is What type of Data is in memory?

Variables: Types of Data

Numbers

Integer
Float
Double
Boolean



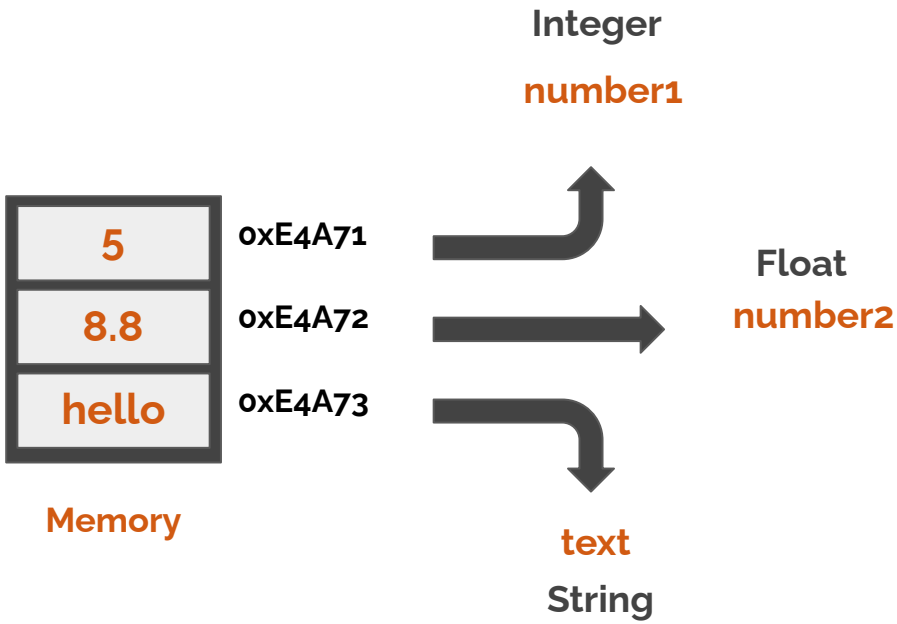
Text

Character
String

Multimedia

Image
Audio
Video

Variables: What Kind of Data Inside



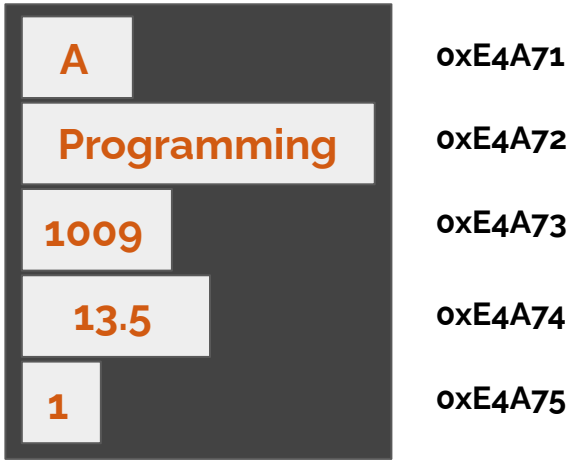
Data Types: Why Inform Memory

Data Types: Why Inform Memory

- To Adjust Size of Allocated Memory Cell
- To Check the Validity of the Operations

Data Types: Size of Memory

Different types of data require Different sizes of cells in memory.



Memory

Data Types: Validity of Operations

We also need to Check whether an Operation applied on the data is Valid or Not.

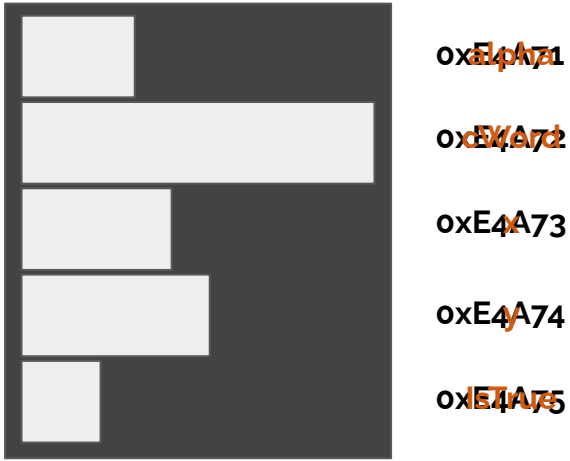
For Example:

- 20 + 20.5

- 20 + Programming

Variable Declaration: Reserve Memory

Reserving the memory location through Variables for certain type of data is also called Variable Declaration.

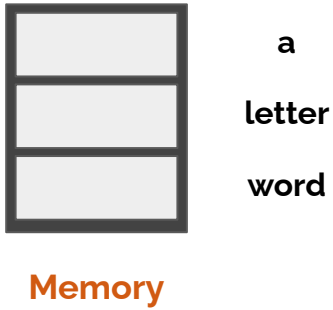


Memory

Variable Declaration: Reserve Memory

In many High Level Languages like C++, Java and C# the variable declaration is done as

```
Datatype nameOfTheVariable;

int a;

char letter;

string word;
```



Uses of Variables

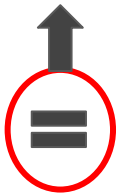
Once the variables are declared and memory is reserved, we can have multiple uses of these variables.

- We can assign values to these variables according to their data types
- We can retrieve values from these variables
- We can apply different mathematical (addition, multiplication, subtraction) and other operations (we will see those in next lecture) on these variables.

Assign values to Variables

We can Assign a value to variable using Assignment Operator.

Assignment Operator



Uses of Variables: Assignment

We can Assign a value to variable using Assignment Operator.

01

Constants

```
int x = 8;
```

02

Variables

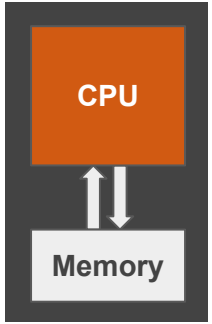
```
int y = x;
```

03

Expressions

```
int z = 5 * 2 / x;
```

Review: CPU Operations



- Some of these operations include
  1. Addition (0010)
  2. Multiplication (0011)
  3. Take Input (1100)
  4. Store Data (1110)
  5. Give Output (0110)
  6. Load Data (0111)

Review: Output

We wrote "Welcome to Programming Fundamentals Class" on Console.

example.cpp - Notepad

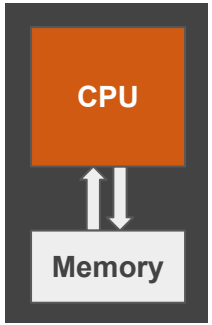
File Edit Format View Help

```
#include <iostream>
using namespace std;

main(){

cout << "Welcome to Programming Fundamentals Class";
}
```

C:\Windows\System32\cmd.exe

```
C:\C++>example.exe
Welcome to Programming Fundamentals Class
C:\C++>
```

CPU Operations



- Some of these operations include
  1. Addition (0010)
  2. Multiplication (0011)
  3. Take Input (1100)
  4. Store Data (1110)
  5. Give Output (0110)
  6. Load Data (0111)

Store Data

Write a C++ program, that **reserves a memory** location of type int and **store** 8 into it.

example.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
using namespace std;
main()
{
    int number;
    number = 8;
}
```

Variable Declaration

Store Data

Write a C++ program, that **reserves a memory** location of type int and **store** 8 into it.

example.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
using namespace std;
main()
{
    int number;
    number = 8;
}
```

Store Data

Similarly, you can **store any type of data** (int, float, string, char)

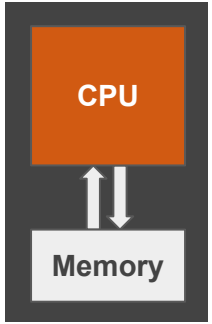
example.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
using namespace std;
main()
{
    int number = 8;
    float decimal = 8.9;
    char letter = 'A';
    string sentence = "This is a string";
}
```

Ln 14, Col 1 | 100% | Windows (CRLF) | UTF-8

CPU Operations



- Some of these operations include
  1. Addition (0010)
  2. Multiplication (0011)
  3. Take Input (1100)
  4. Store Data (1110)
  5. Give Output (0110)
  6. Load Data (0111)

Store Data and Give Output

Write a C++ program, that reserves a memory location of type int and store 8 into it and display the value of variable on screen.

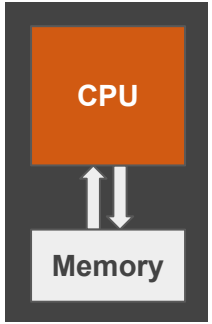
*example.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
using namespace std;
main()
{
    int number;
    number = 8;
    cout << number;
}
```

When we display a variable on the console then we do not use double quotes (“ ”)

CPU Operations



- Some of these operations include
  1. Addition (0010)
  2. Multiplication (0011)
  3. Take Input (1100)
  4. Store Data (1110)
  5. Give Output (0110)
  6. Load Data (0111)

Take Input

Computers can **take input** in different forms using:

Keyboard

Mouse

Microphone

Take input from Console

Write a C++ program, that takes name as input from the console and then show it with a message on the console.

C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\C++>c++ example.cpp -o example.exe

C:\C++>
```

Input from the Console in C++

Variable declaration means CPU is allocating some space in memory for specific type of data (int, float, string, char)

example.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
using namespace std;
main()
{
    string user_name;
    cout << "Please Enter your Name: ";
    cin >> user_name;
    cout << "User Entered " << user_name << " as his/her name.";
}
```

Ln 14, Col 1 | 100% | Windows (CRLF) | UTF-8

We have to store the input in memory, therefore, we have reserved memory for **string** type of data

Input from the Console in C++

cout command is used to display output on Console in C++.

example.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
using namespace std;
main()
{
    string user_name;
    cout << "Please Enter your Name: ";
    cin >> user_name;
    cout << "User Entered " << user_name << " as his/her name.";
}
```

Ln 14, Col 1 | 100% | Windows (CRLF) | UTF-8

We display a message to user, so he knows which type of input he has to enter.

Input from the Console in C++

cin command is used to take input from the Console in C++.

example.cpp - Notepad

File Edit Format View Help

cin stands for Character Input.

```
#include <iostream>
using namespace std;
main()
{
    string user_name;
    cout << "Please Enter your Name: ";
    cin >> user_name;
    cout << "User Entered " << user_name << " as his/her name.";
}
```

Ln 14, Col 1 | 100% | Windows (CRLF) | UTF-8

Input from the Console in C++

cin command is used to take input from the Console in C++.

example.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
using namespace std;
main()
{
    string user_name;
    cout << "Please Enter your Name: ";
    cin >> user_name;
    cout << "User Entered " << user_name << " as his/her name.";
}
```

Ln 14, Col 1 | 100% | Windows (CRLF) | UTF-8

cin is a predefined command that reads data from the keyboard with the extraction operator (>>)

Input from the Console in C++

cout command is used to display output on Console in C++.

example.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
using namespace std;
main()
{
    string user_name;
    cout << "Please Enter your Name: ";
    cin >> user_name;
    cout << "User Entered " << user_name << " as his/her name.";
}
```

Ln 14, Col 1 | 100% | Windows (CRLF) | UTF-8

Data in the variable is displayed on the console without using the double quotes (“ ”)

Input from the Console in C++

Output on the console of the program is as follows:

```
C:\C++>c++ example.cpp -o example.exe

C:\C++>example.exe
Please Enter your Name: Talha
User Entered Talha as his/her name.
C:\C++>
```

Input from the Console in C++

Similarly, you can take any type of data (int, float, string, char) as input from the console.

example.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
using namespace std;
main()
{
    int number;
    cin >> number;
}
```

example.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
using namespace std;
main()
{
    float number;
    cin >> number;
}
```

example.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
using namespace std;
main()
{
    char alphabet;
    cin >> alphabet;
}
```

Learning Objective

Explain why we need Variables, what is their Relation with the Memory, what is a Data Type, why we need it, what is its Role in Variable Declaration, how to use Variables while taking input and giving output at screen.

Conclusion

- Variable is a Human Friendly name of the Memory Location.
- Data can be of the following 3 types.
  a. Number
  b. Text
  c. Multimedia
- Telling the memory about the Datatype helps
  a. To Adjust Size of Allocated Memory Cell
  b. To Check the Validity of the Operations
- Variable Declaration means Reserving the memory location through Variables for certain type of data.

Self Assessment

1. What is a Variable?
2. How we can store and load data from the Memory using variables?
3. From the given table below, tell which Variable Names are Valid and which are not.

Variable	Valid/Invalid
mul*	
Foo	
Do it	

Self Assessment

4. Define Variable Declaration. And Declare a variable to store a value of 58.9

5. Write the Datatypes of the following data given in the table

Data	Datatype
400.6	
My name is Kaka	
C	
12	

6. Declare the variables to store the above mentioned data in the variables.

Hint: float a; (a is a variable that will store float type of data)