CS F111: Computer Programming

(Second Semester 2021-22)

Lect 4: Variables & Algorithm

Dr. Nikumani Choudhury

Asst. Prof., Dept. of Computer Sc. & Information Systems nikumani@hyderabad.bits-pilani.ac.in

BITS Pilani

Hyderabad Campus

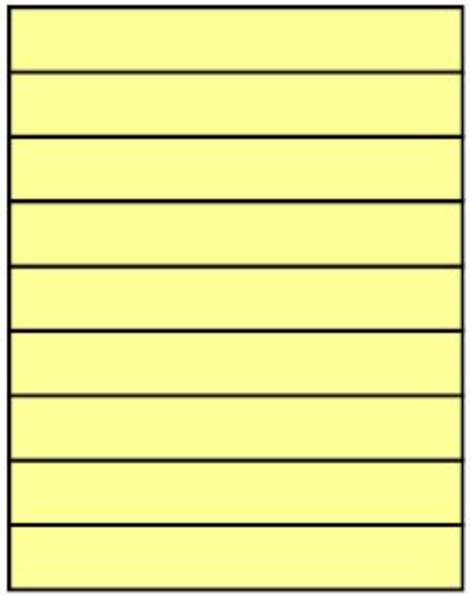
Variables and Constants

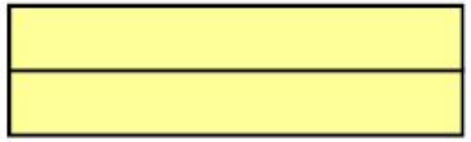
- **Most important concept for problem solving using computers.**

- **All temporary results are stored in terms of variables**
  - **The value of a variable can be changed.**
  - **The value of a constant do not change.**

- **Where are they stored ?**
  - **In main memory.**

Memory map



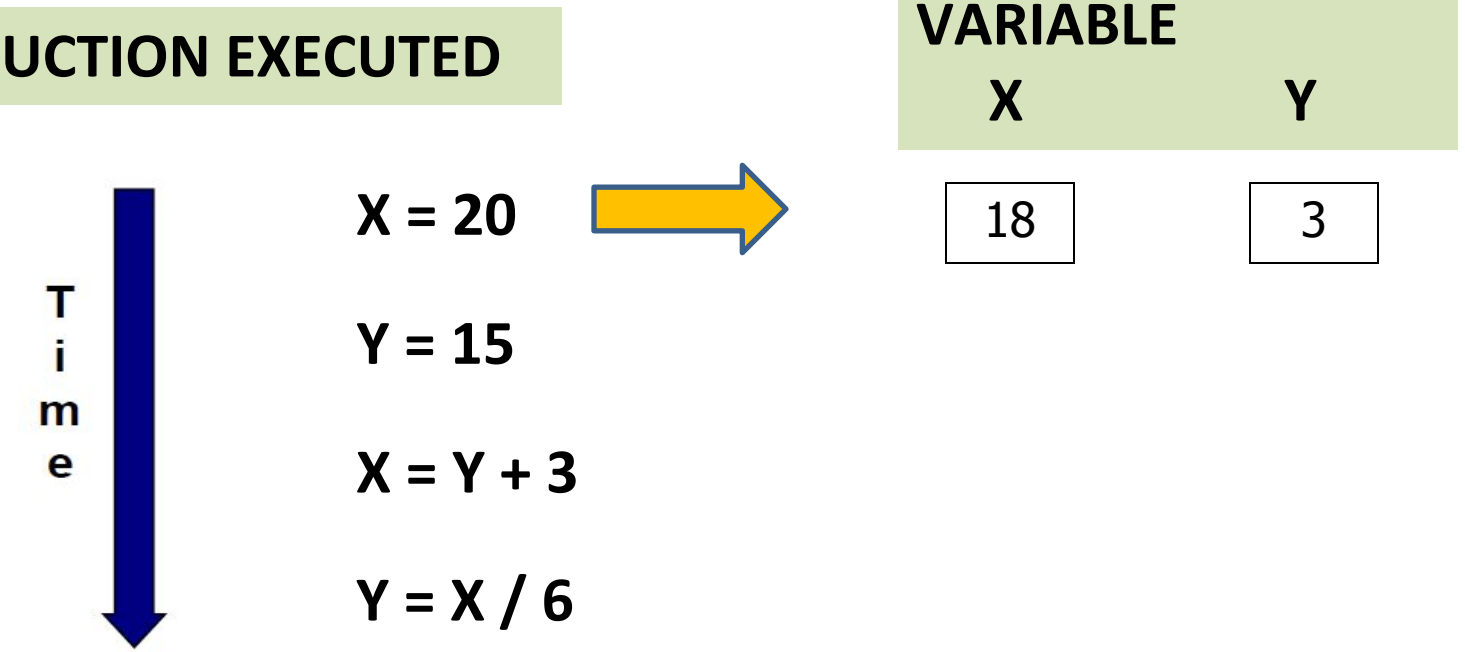


- list of storage locations
- every variable is mapped to a particular memory address

Variables in Memory

INSTRUCTION EXECUTED	VARIABLE X
X = 10	X 21
X = 20	
X = X + 1	

Variables in Memory (contd.)



Contd.

- How does memory look like (logically)?
  - As a list of storage locations, each having a unique address.
  - Variables and constants are stored in these storage locations.
  - A variable is like a *bin*
- The contents of the *bin* is the *value* of the variable
- The variable name is used to refer to the value of the variable
- A variable is mapped to a *location* of the memory, called its *address*.

Data Types in C

Data Type	Description	Bytes (Typically)	Range
int	Stores integer	4	-2147483648 to 2147483647
char	Stores single character	1	-128 to 127
float	Stores real number	4	-3.4E+38 to +3.4E+38
double	Stores real number but with more precision than float	8	-1.7E+308 to +1.7E+308
void	Associated with no data type	-	

Depends on the compiler & its abstraction implementation

```c
#include <stdio.h>
int main()
{
    int a = 1;
    short int e = 1;
    long int d = 1;
    char b = 'G';
    double c = 3.14;

    // printing the variables defined above along with their sizes
    printf("Hello! I am a character. My value is %c and my size is %lu byte.\n", b, sizeof(char));
    // can use sizeof(b) above as well

    printf("Hello! I am an integer. My value is %d and my size is %lu bytes.\n", a, sizeof(int));
    printf("Hello! I am a short integer. My value is %hd and my size is %lu bytes.\n", e, sizeof( short int));
    printf("Hello! I am a long integer. My value is %ld and my size is %lu bytes.\n", d, sizeof( long int));
    // can use sizeof(a) above as well

    printf("Hello! I am a double floating point variable. My value is %lf and my size is %lu bytes.\n", c, sizeof(double));
    // can use sizeof(c) above as well
    return 0;
}
```

input

```
Hello! I am a character. My value is G and my size is 1 byte.
Hello! I am an integer. My value is 1 and my size is 4 bytes.
Hello! I am a short integer. My value is 1 and my size is 2 bytes.
Hello! I am a long integer. My value is 1 and my size is 8 bytes.
Hello! I am a double floating point variable. My value is 3.140000 and my size is 8 bytes.


...Program finished with exit code 0
Press ENTER to exit console.
```

Data Type Qualifiers

- **Usage : qualifier data_type**
- **Types of qualifiers:**

  short
  long
  signed
  unsigned

Examples: **(ShortHand)**

1. **unsigned int ⇔ unsigned**
2. **short int ⇔ short (typically 2 bytes)**
3. **long int ⇔ long (typically 4 bytes)**

Types of variable

- We must *declare the type of every variable we use in C.*

- Every variable has a *type (e.g. int) and a name.*

- Declarations of types should always done before use

1. int
2. char
3. float
4. double

Declaration of Variables

There are two purposes:

1. It tells the compiler what the variable name is.
2. It specifies what type of data the variable will hold.

General syntax:

data-type variable-list;

- Examples:
- int salary, bonus;
- int x, y, z;
- float simple_interest;
- char ch, option;

How to print to screen ???

printf(): printf(formatString, list of variables);

- It requires a format string in which we can specify the text to be printed out

  printf ("Simple Interest is %d for this year\n", si);

- The format specifier %d works as a placeholder and is to be embedded in the output as a decimal number in place of %d.

Format Specifier

Data Type	Format Specifier
int	%d
char	%c
float	%f
double	%lf
unsigned int	%u
long int	%ld or %li
long long int	%lld or %lli
long double	%Lf or %LF

How to take Input from keyboard ???

scanf(): scanf(formatString, list of variables);

- formatString contains only format specifier of the variables
- List of variables are separated by commas and should be address of the variable and hence generally have & before variable name.

```
scanf("%d", &num);
scanf("%f", &si);
scanf("%d %f", &amount, &rate);
```