

CS 101 Computer Programming and Utilization

Lecture 3

Variables, Values, Assignment and Types

Jan 14 Friday 11:05-12:30 PCS D2

Jan 14 Friday 2:00-3:30 FCK D4

Prof. R K Joshi

Computer Science and Engineering

IIT Bombay

Email: rkj@cse.iitb.ac.in

Revision: Elements of a Program

- Main Procedure
- Procedure body
- Return statement
- Exit Status
- Statements
- Sequence of statements
- Keywords
- Identifiers
- Operators
- Syntax
- Meaning
- Grammar
- Libraries
- Input and Output (to be continued..)

Cout and cin

- Predefined identifiers
- Defined in library *iostream*
 - in namespace *std*
 - *namespaces are used to organize entities*
 - Cin and cout are names of input and output stream objects
 - These objects are connected to input and output devices
- Operators << and >> can be used to perform input/output operations these objects. .
- Reading from cin: `cin >> x;`
- Writing to cout: `cout << x;`

(demo: linking input with output..try commenting the using namespace statement)

A Variable

- Variables hold objects used in the program
- A value can be assigned to a variable
- The assigned values can be changed through reassignment
- Variables occupy specific memory locations in the running program
- Assignment to a variable changes the content of its corresponding location

Assigning Values to Variables

- An example of Integer Values

```
int i ;
```

```
i = 212 ;
```

Let's execute this program step by step..

```
int main () {  
int x;  
    x = 10;  
    cout << x * x ;  
    x = 20;  
    cout << x * x ;  
    return 0;  
}
```

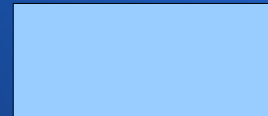
Memory Location for holding the value of x

```
int main () {
```

```
int x;
```



x



```
    x = 10;
```

```
    cout << x * x ;
```

```
    x = 20;
```

```
    cout << x * x ;
```

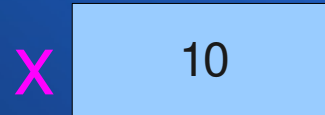
```
    return 0;
```

```
}
```

An assignment has been made

```
int main () {
```

```
int x;
```



```
    x = 10; ←
```

```
    cout << x * x ;
```

```
    x = 20;
```

```
    cout << x * x ;
```

```
    return 0;
```

```
}
```


Expression evaluated first and then sent to cout

```
int main () {
```

```
int x;
```

x 10

```
    x = 10;
```

```
    cout << x * x ;
```

```
    x = 20;
```

```
    cout << x * x ;
```

```
    return 0;
```

```
}
```

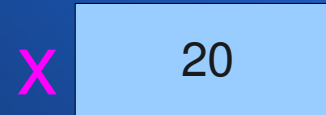
Output

100

Another assignment..the object remains but the previous value wiped out ..

```
int main () {
```

```
int x;
```



```
    x = 10;
```

```
    cout << x * x ;
```

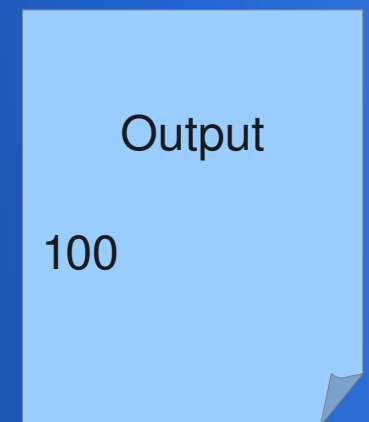
```
    x = 20;
```



```
    cout << x * x ;
```

```
    return 0;
```

```
}
```



The expression evaluated again.. latest value of x is picked up..

The result of the expression sent to cout

```
int main () {
```

```
int x;
```

x 20

```
    x = 10;
```

```
    cout << x * x ;
```

```
    x = 20;
```

```
    cout << x * x ;
```



```
    return 0;
```

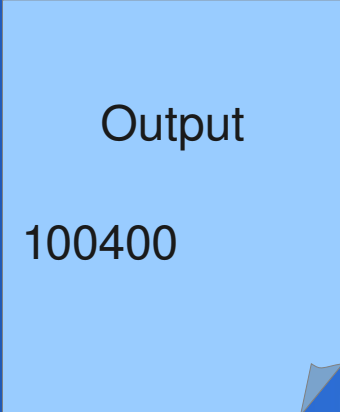
```
}
```

Output

100400

Main procedure is completed.. memory allocated to this procedure is returned back to operating system... you may still see the output on screen.

```
int main () {  
    int x;  
    x = 10;  
    cout << x * x ;  
    x = 20;  
    cout << x * x ;  
    return 0;  
} ←
```



Output
100400

Types

- In a program, we need values of various kinds
 - ..-3, -2, -1, 0, 1, 2, 3,
 - 'A', 'B', 'C', ..., 'a', 'b', 'c', 'd', ..
'1', '2', '3', ...
'!', '@', '<',
 - 12.34, 1.234, 3.142, ..
 - “Robin Hood”
 - { “Robin Hood”, 'M', 15, ”Mumbai” }

Restricting the set of values assignable to a given variable

- Declaring a Type of every variable
- A value of the declared type can be assigned to the variable
- A value not of the declared type cannot be assigned ----> “Typing Error” (demo)

Use of Types

- Types are a means of organization of data
- Types are used to determine storage requirement for variables
- They are also used in “type checking”
 - Check the Type of the value being assigned
- Standardized types enable easy exchange and use of stored information

Variable Declarations

```
int main () {  
    int x;          <----- A variable of type int  
    int x, y, z;    <----- Many variables of type int  
    int x = 10;    <-- An initialized variable of type int  
  
}
```


The Assignment Statement

<code>int x, y;</code>	variables x and y of type int are declared
<code> x = 124;</code>	variable x is on the left hand side
<code> y = 100;</code>	variable y is on the left hand side
<code> y = x;</code>	variable x is on the right hand side
	Variable y is on the left hand side

lvalue and rvalue

- Values that occur on the left hand side and on the right hand side of an assignment
- In expression $e1 = e2$, $e1$ is lvalue and $e2$ is rvalue
- Is 30 a valid lvalue?
 - Try it out demo
- Location vs. value contained in the location

Using expressions in assignment statements

- $x = x + 1 ;$
- $x = x + y ;$
- $x = 2 * x ;$
- $x = (2 * x) + (2 * x) ;$

demo

Sizes of variables

- Binary Numbers
- Bit: 0s and 1s
 - Why do computers need to use binary numbers?
- A Byte: 8 bits
 - How many values?
- A variable with 2 Bytes storage:
 - how many values can it hold?
- For encoding all English characters, how many bits do you need?

How much storage for a variable?

Define a new type called Mountain

Given that it has 8 possible values all in all,

How much memory should a variable of type Mountain occupy?

Some Standard Types and their sizes

- char 1 byte
- int 4 bytes
- short int 2 bytes
- float 4 bytes
- double 8 bytes
- bool 1 byte

Demo

Standard Types Vs. User Defined Types

- Commonly used types are predefined in the language
- They are added as keywords in the language
- Int, float, char, ..
- New types can be created by the user
 - composites from existing types
 - e.g. a record of a person's details