CS 101 Computer Programming and Utilization

Lecture 2
Basic Elements of a Program

Tue Jan 10, 2010 2:00-3:30, FCK Auditorium
Wed Jan 11, 2010 11:00-12:30 PCS Auditorium

Prof. R K Joshi
Computer Science and Engineering
IIT Bombay
Email: rkj@cse.iitb.ac.in

Revision: Programs

- A program represents knowledge about doing something, and a machine stores this knowledge and applies it to a given situation.
- Input, Output and Logic
- Handling possible errors, problematic conditions
- State: A Programmed System may remember something from earlier interactions, and consider it as one input in the next interaction.
    - as in Remote controls, telephones, vending machines

A Program that does Nothing!

```
int main () {

    return 0;
}
```

main

```
int main () {
    return 0;
}
```

Main:

- It's the name of a procedure
- The name is pre-recognized in C++
- It indicates the entry point for code execution

Value returned by main

```
int main () {
    return 0;
}
```

- The return value represents exit status of the program
- After the program terminates, the status can be inspected at the command prompt (echo $?)
- Use value 0 to indicate correct execution
- Use other values for returning error codes

Logic of the Procedure { .... }

```
int main () {
    return 0;
}
```

- The logic is embedded inside curly brackets
- After the logic portion, the return statement appears.
- So, what logic does the above program implement?

Inputs to the procedure ()

```
int main () {
    return 0;
}
```

- Parenthesis are used to provide values as inputs to the procedure
- In the above program, we are not expecting any input
- For main function, the input values will come directly from command line
- We will come back to this point later ...

Statements

```
int main () {
    return 0;
}
```

- A procedure body is composed of a sequence of statements
- Statements are separated by semicolons ;
- There are various types of statements. They have syntactic rules (grammar)
- Syntax of the Return statement: return expression ;

Syntax

- Program should follow the syntax rules of the language used for programming
- These rules make the grammar of the language
- Syntax rules are provided for all types of programming constructions possible in the language
- Meanings are associated with syntactic forms
  - just like how we understand a paragraph in a natural language
  - Use of keywords, symbols, identifiers for names, statements, sequence of statements..

Keywords

```
int main () {
    return 0;
}
```

- Keywords are reserved words
- They cannot be redefined by the programmer
  - Which means they can't be used to name objects
- Just like dictionary words in a natural language
- Int, return are keywords

Take a look at Some Keywords

int	float	char
if	else	do
while	for	return
switch	case	const

Operators

Symbols to specify predefined operations on values

=	(assignment)	+	(addition)
-	(subtraction)	/	(division)
*	(multiplication)	%	(modulus)

There are many other operators of different kinds

Identifiers

- Names of values, objects
- 'main' is an identifier
  - But it's predefined!
- Examples: a, b, c, i, x, no_of_students, total, average_value, max, count, current, age, name, weight, force, balance...
- A programmer can create these names in a program and use them in the program.

More Programs

- Over to Demos

Cout and cin

- Predefined identifiers
- Names of input and output objects
- Operators << and >> can be used to operate these objects
- Reading from cin
- Writing to cout

Use of Existing Libraries

- A library contains pre-written code and predefined entities
- It's organized in terms of namespaces
  - Just like how central library is organized in terms of sections
- The include statement
- The *using* statement