Fundamental Data Types

Yaping Jing

CS270 – Computer Science II

Question

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

int main() {
    int num;
    cout << "Enter an integer number"<< endl;
    cin >> num;
    cout << "the number you entered is: " << num << endl;

    return 0;
}
```

Which are the library identifiers in this program?

Question

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

int main() {
    int 3_num;
    cout << "Enter an integer number: "<< endl;
    cin >> num;
    cout << "the number you entered is:  " << num << endl;

    return 0;
}
```

Is there error(s) in this program?

Question

C++ requires that all variables used in a program be given a data type.

true or **false**?

Data Types – Integers

```
int  num ;

short int small_num ;

long int large_num ;

unsigned long int large_num ;
```

Question

Given that a **char** costs 1 byte, what is the range of (unsigned) integer values that a character can represent?

signed?

Given that a **short int** costs 2 byte, what is the range of (unsigned) integer values that a **short int** can represent?

signed?

Question

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

int main() {
    unsigned int num = -2;
    cout << num << endl;

    return 0;
}
```

Will this program compile? what's the contents of num after execution? Try this out on your computer!

Data Types – Real Numbers

```
double  probability ;

float  account_balance = 3.14E3 ;
```

Example

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

int main() {
    float x ;
    cout << "Enter a real number: "<< endl;
    cin >> x;
    cout << "the number you entered is:  " << x << endl;

    return 0;
}
```

Check the Size

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

int main() {
  cout << sizeof(int) << endl;

  cout << sizeof(short) << endl;

  cout << sizeof(double) << endl;

  cout << sizeof(long) << endl;

  return 0;
}
```

Type Casting

In C++, the integer division $5/2$ evaluates 2.
What do you do if you want to get the answer of 2.5?
Add a decimal point and zero to one or both numbers
e.g.

$$5.0/2,$$

$$5/2.0,$$

$$5.0/2.0$$

C++ Syntax - Type Casting (Cont')

What if both the numerator and the divisor are variables?

Use a type cast.

e.g.

answer = static_cast<double> (numerator) / denominator

or

answer = static_cast<int> (3.58) gives an integer value of 3.

Question

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

int main() {
    int x, y;
    double z;

    x = 7;
    y = 2;
    z = 7.0;
    cout << "answer1=" << x/y << endl;
    cout << "answer2=" << z/y << endl;
    cout << "answer3=" << static_cast<double>(x)/y << endl;

    return 0;
}
```

Characters

```
char ch;
ch = 'a'
```

Example

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

int main() {
    char  ch1 = 'a';
    cout << "ch1 is " << ch1  << endl;

    return 0;
}
```

Special Character Literals

- `'\n'` newline
- `'\b'` backspace
- `'\\'` backslash
- `'\t'` tab
- `'\0'` null
- `'\"'` double quote
- `'\''` single quote

Example

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

int main()
{
  cout << "hello\nWorld\n" << endl;

  cout << "this\tis\ta\tgood\n" << endl;
  return 0;
}
```

Print Escape Character Example

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

int main() {
  cout << "␣\\n␣" << "␣=␣" << (int)'\n' << endl;

  cout << "␣\\t␣" << "␣=␣" << (int)'\t' << endl;

  return 0;
}
```

Constants

- Variables can be ”initialized” at the same time when a variable is declared.
- e.g.

```
double PI = 3.1415926;
```

- If preceded with “const”, variables value can not be altered during the execution of a program.
- e.g.

```
const double PI = 3.1415926;
```

Example

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

const double PI = 3.1415926;

int main() {
  PI = 9.666;
  cout << "x = " << x << endl;
  return 0;
}
```

What will be the error here?

Enumerations

Constants of type int may also be declared with an “enumeration” statement.

e.g.

enum weekdays {MON, TUES, WED, THURS, FRI };

is shorthand for

```
const int MON = 0;
const int TUES = 1;
const int WED = 2;
const int THURS = 3;
const int FRI = 4;
```

Enumerations

enum {MON=1, TUES, WED, THURS=-1, FRI };

if TUES = 2. FRI=?

Enum – User Defined Data Type

```
enum    WEEKDAYS    {MO, TU, WE, TH, FR};

        WEEKDAYS  wd;
```

Another Example Using Enum

```
#include <iostream>
using namespace std;
enum WEEKDAYS {MO, TU, WE, TH, FR};
void getActivity(WEEKDAYS wd){
    switch(wd) {
        case MO:
            cout << "we␣have␣a␣group␣meeting.␣" << endl;
            break;
        case TU:
          cout << "we␣go␣to␣swimming.␣" << endl;
          break;
        default:
            cout << "we␣go␣to␣study.␣" << endl;
          break;
     }
}
int main(){
  getActivity(MO);  getActivity(FR);
  return 0;
}
```

Type bool

```
bool  foo = true;
```

```
bool  bar = false;
```

Variable Declarations

- Variables have to be declared before they can be used in a program. E.g.

```
float number;

number = 0.666;

ch c = 'z';

char[] str = "hello";
```

Arithmetic Operations

Symbol	Operation	Example	Value
+	Addition	3+5	8
-	Subtraction	5-2	3
*	Multiplication	3 * 7	21
/	Division	9/2	4
%	Modulus	10%4	2

Shorthand Arithmetic Assignment Statements

Example	Equivalent to
num +=1;	number = num + 1;
total -= promotion;	total = total - promotion;
num *=2;	number = num * 2;

Shorthand Arithmetic Assignment Statements

Example	Equivalent to
num +=1;	number = num + 1;
total -= promotion;	total = total - promotion;
num *=2;	number = num * 2;

Can you come up with example expressions for operators / and %?

More Shorthand Arithmetic Expressions

```
num = num+1;
```

can also be written as:

```
num++;
```

or

```
++ num;
```

Comparison Operations

$<$	less than	$2 < 6$	TRUE
$<=$	less than or equal to	$6 <= 2$	FALSE
$>$	greater than	$2 > 9$	FALSE
$>=$	greater than or equal to	$8 >= 7$	TRUE
$==$	equal to	10 == 3	FALSE
$!=$	not equal to	10 != 2	TRUE

Operator Precedence

Type	**Operators**
scope resolution	namespace name :: member
postfix operator	var++ var–
prefix operator	++var –var
multiplication/division	* / %
addition/subtraction	+ -
comparison	< > = <=
equality	== !=
logical and	&&
logical or	||
assignment	= += -= *= /= %=

Branches: If – Else Statements

if (condition)
    statement;

else if (condition)
    statement;

else
    statement;

Note that **else if** and **else** parts are optional.

If-else Example

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

int main()
{
    int num;
    cout << "␣enter␣a␣number␣" << endl;
    cin>>num;
    if ( num > 0)
          cout << "num␣is␣positive␣" << endl;
    else
        cout << "num␣is␣not␣positive␣" << endl;

    return 0;
}
```

Branches: Switch Statement Syntax

```
switch (variable) {
    case  (some value in variable) :
        do some operation;
        break;
    case  (some value in variable) :
        do some operation;
        break;
        ...
    default :
        do some operation;
        break;
}
```

Switch Statement Example

```
char command;
cin >> command;

switch(command) {
    case  'S':
          cout<<sizeof(short)<<endl;
          break;
    case  'D':
          cout<<sizeof(double)<<endl;
          break;
    default:
      cout << "unrecognized command " << endl;
      break;
  }
```