# Tutorial 1: Abstract Data Types and C++ Basics

1. What is an Abstract Data Type (ADT) ?

The following exercises are about some basic features of C++. If you need help, please refer to Appendix C (Basic C++) of Nyhoff’s textbook.

1. State whether each of the following is *true* or *false*. If false, explain why.
   1. C++ considers the variables **number** and **NuMber** to be identical.
   2. The modulus operators (%) can be used only with integer operands.
   3. The arithmetic operators \*, /, %, +, and – have the same level of precedence.
   4. The **default** case is required in the **switch** selection structure.
   5. The address operator & can be applied only to constants and to expressions.
   6. A pointer that is declared to be of type void can be dereferenced.
2. Write a single C++ statement to accomplish each of the following : (assume that no **using** statements have been used)
   1. Declare a variable **number** to be of type **int**.
   2. Prompt the user to enter an integer, end your prompting message with a colon (:) followed by a space and leave the cursor positioned after the space.
   3. Read an integer from the user at the keyboard and store the value entered in integer variable **number**.
   4. If the variable **number** is not equal to 7, print “The variable number is not equal to 7”.
   5. Print the message “This is a C++ program” on one line.
   6. Print the message “This is a C++ program” on two lines, in which the first line ends with C++.
   7. Print the message “This is a C++ program” with each word separated from the next by a tab.
3. Write a C++ program that calculates and prints the sum of the integers from 1 to 10. Use the while-loop structure to loop through the calculation and increment statements.
4. Find errors (including logic errors) in each of the following code segments and explain how to correct them.
   1. x = 1;

while (x <= 10);

x++;

}

* 1. switch (n) {

case 1:

cout << “The number is 1” << endl;

case 2:

cout << “The number is 2” << endl;

break;

default :

cout << “The number is not 1 or 2” << endl;

break;

}

* 1. The following code should print the values 1 to 10.

n = 1;

while ( n < 10)

cout << n++ << endl;

1. Write a C++ program containing only one function, main( ) function, that performs the following specified operations:
2. Declare two double variables **number1** and **number2** and **number1** has been initialized to 7.3.
3. Declare a variable **fPtr** to be a pointer of type **double**.
4. Assign the address of variable **number1** to pointer variable **fPtr**.
5. Print the value stored at location pointed to by **fPtr**.
6. Assign the value stored at location pointed to by **fPtr** to variable **number2**.
7. Print the value of **number2**.
8. Print the address of **number1**.
9. Print the address stored in **fPtr**. Is the value printed the same as the address of number1?
10. What is the output of the following C++ code?
    1. int x, y, \*p = &x, \*q = &y;

\*p = 35;

\*q = 98;

\*p = \*q;

cout << x << “ ” << y << endl;

cout << \*p << “ ” << \*q << endl;

* 1. int x, y, \*p = &x, \*q = &y;

x = 35;

y = 46;

p = q;

\*p = 78;

cout << x << “ ” << y << endl;

cout << \*p << “ ” << \*q << endl;

1. Write a simple calculator program that repeatedly allows the user to select +, -, \*, or / from a menu of operations on real values and then enter the two operands, and then computes the result of applying the selected operation to those operands. However, **the only named variables you may use are pointer variables**, all others must be anonymous variables.