Third School on LHC Physics

C++ Handouts

Prepared by: Taimoor Khurshid

Use of cout:

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello ";
    cout << "World.\n";
    cout << "Hello World.\n";
    int a = 5;
    cout << "a = ";
    cout << a;
    cout << "\n";
    int b = 23;
    cout << "b = " << b << "\n";
}
```

Input/Output

```
#include <iostream>
using namespace std;
int main() {
    int year, age;
    cout << "What year were you born? ";
    cin >> year;
    age = 2002 - year;
    cout << "You are " << age << " years old.\n";
}
```

Use of string:

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    cout << "Please enter your name: ";
    string name;
    cin >> name;
    cout << "Your name is " << name << "\n";
}
```

Use of getline:

```
#include <iostream>
```

```
#include <string>
using namespace std;
int main() {
    cout << "Please enter your name: ";
    string name;
    getline(cin, name);
    cout << "Your name is " << name << "\n";
}
```

Use of if Statement

```
#include <iostream>
using namespace std;
int main() {
    cout << "Enter your age: ";
    int age;
    cin >> age;
    if (age < 20)
        cout << "You are still young!\n";
}
```

Use of if-else statement:

```
#include <iostream>
using namespace std;
int main() {
    cout << "Enter your age: ";
    int age;
    cin >> age;
    if (age < 20)
        cout << "You are still young!\n";
    else
        cout << "You are not so young anymore.\n";
}
```

Simple Program to understand switch statement:

```
#include <iostream>
using namespace std;
int main() {
    cout << "What would you like to do:\n";
    cout << "1. Add an entry to the address book\n";
    cout << "2. Look up an address\n";
    cout << "3. Remove an entry\n\n";
    cout << "Your choice: ";
    int selection;
    cin >> selection;
    switch (selection) {
    case 1:
        cout << "Sorry, this feature has yet to be "
             << "programmed.\n";
        break;
    case 2:
        cout << "Sorry, this feature was not yet "
             << "implemented.\n";
        break;
    case 3:
        cout << "Access denied.\n";
        break;
    default:
        cout << "You can't do that. You must choose "
             << "1, 2 or 3.\n";
        break;
    }
}
```

Write a program which takes the input of two numbers and asks you whether you want to multiply or add them: Use if else statement.

```
#include <iostream>
using namespace std;
int main() {
    cout << "1. Calculate a sum\n";
    cout << "2. Calculate a product\n";
    cout << "Your choice: ";
    int selection;
    cin >> selection;
    if (selection == 1) {
        cout << "Enter the two numbers to sum "
             << "separated by a space: ";
        int first, second;
        // Notice how cin can be used to input more
        // than one value in a single call
        cin >> first >> second;
        cout << "The sum is: " << first + second
             << "\n";
    }
    else {
        cout << "Enter the two numbers to multiply "
             << "separated by a space: ";
        int first, second;
        cin >> first >> second;
        cout << "The product is: " << first * second
            << "\n";
    }
}
```

Write a program which calculates Area of square and circle, by giving you choice of which one to calculate. Use Switch Statement:

```
 #include <iostream>
using namespace std;
int main() {
    // First display the menu.
    cout << "What do you want to do:\n";
    cout << "1. Calculate the area of a square\n";
    cout << "2. Calculate the area of a circle\n";
    cout << "Your choice: ";
    // Take the user's selection.
    int choice;
    cin >> choice;
    // Do the right thing based on the user's
    // selection.
    switch (choice) {
    case 1: {
        // We need braces here because otherwise we
        // cannot declare the variable 'side' below.
        cout << "Please enter the side length: ";
        double side;
        cin >> side;
        // We do not accept negative side lengths.
        if (side < 0)
            cout << "There can be no squares with "
                << "negative side lengths. Bye.\n";
        else
            cout << "The area is " << side * side
                << ".\n";
        break;
    }

    case 2: {
        // This is all the same as above, just for
        // circles.
        cout << "Please enter the radius: ";
        double radius;
        cin >> radius;
        if (radius < 0)
```

```
            cout << "There are no circles with "
                << "negative radiuses. See you.\n";
        else
            cout << "The area is "
                << radius * radius * 3.1415926
                << ".\n";
        break;
    }
    default:
        // The user entered an invalid selection;
        // display an error message.
        cout << "Your selection isn't valid.\n";
        break;
    }
}
```

Generate numbers from 1 to 9 by using while loop

```
#include <iostream>
using namespace std;
int main() {
    int i = 0;
    // Execute the loop until (i < 10) is not true
    // anymore, in other words until i >= 10
    while (i < 10) {
        // Print the number
        cout << i << " ";
        // Increase i by one
        i = i + 1;
    }
}
```

Use of do-while loop

```
#include <iostream>
using namespace std;
int main() {
    int i;
    // Keep asking until the number is greater than 10
    do {
        cout << "Please enter a number greater than "
          << "10:";
        cin >> i;
    } while (i > 10);
    cout << "You entered: " << i << "\n";
}
```

Use of for loop: Produce integers from 0 to 9 by using for loop

```
#include <iostream>
using namespace std;
int main() {
   for (int i = 0; i < 10; ++i) {
      cout << i << " ";
   }
}
```

Break and Continue Statements:

Let the user enter 100 positive numbers, or a negative number

```
#include <iostream>
using namespace std;
int main() {
   cout << "Enter 100 positive numbers, or a "
       <<" negative number to abort.\n";
   // Notice that here we declare i *outside* of the
   // loop. You'll see later why.
   int i;
   // Here we start counting at 1 and not at 0 because
   // otherwise the program would ask for number #0,
   // then for number #1, but we want it to start at
   // 1.
   for (i = 1; i <= 100; ++i) {
      cout << "Enter the number #" << i << ": ";
      int n;
      cin >> n;
      if (n < 0)
         break;
   }

   // If we hadn't declared i outside the loop, we
   // couldn't access it here because it'd be out of
   // scope.
   if (i == 100)
      cout << "You are a real man.\n";
   else
      cout << "You stopped after " << i
          << " numbers, coward!\n";
}
```

Print all numbers from 0 to 100 except the multiples of 7.

```
#include <iostream>
```

```
using namespace std;
int main() {
    for (int i = 0; i < 100; ++i) {
        // Skip all the multiples of 7
        // Operator % calculates the modulo (remainder)
        // of a division; if (i % 7) is equal to zero
        // this means that i is a multiple of 7.
        if ((i % 7) == 0)
            continue;
        cout << i << " ";
    }
}
```

Nested Loops:

This program prints out a multiplication table as follows:

```
1  2  3
2  4  6
3  6  9
```

You could use more numbers than just from one to three, but then there'd be problems with the alignment (because there'd be numbers with one and with two digits).

```
#include <iostream>
using namespace std;
int main() {
    for (int y = 1; y <= 3; ++y) {
        for (int x = 1; x <= 3; ++x) {
            cout << x * y << " ";
        }
        cout << "\n";
    }
}
```

Following program calculates factorials. (The factorial of a positive integer number n is mathematically written as n! and is the product of all numbers from 1 to n;
Mathematically
n! = 1 x 2 x … x n
7! = 1 x 2 x 3 x 4 x 5 x 6 x 7 = 5040

```
#include <iostream>
```

```
using namespace std;
int main() {
    cout << "This program calculates the factorial "
         << "of n.\n";
    cout << "Please enter n: ";
    int n;
    cin >> n;
    if (n < 0)
        cout << "n must be non-negative.\n";
    else {
        int factorial = 1;

        for (int i = 1; i <= n; ++i)
            factorial = factorial * i;
        cout << n << "! = " << factorial << "\n";
    }
}
```

Use of Functions:

```
#include <iostream>
using namespace std;
int square(int n) {
    return n * n;}
int main() {
    int num;
    do {
        cout << "What do you want to know the square "
             << "of (enter 0 to quit): ";
        cin >> num;
        cout << "The square of " << num << " is "
             << square(num) << ".\n";
    } while (num != 0);
}
```

Examples of some functions

1.// A silly function which always returns 3

```
int three() {
    return 3;
}
```

2. // Calculates the average of 2 numbers

```
double average(double n1, double n2) {
    return (n1 + n2) / 2;
}
```

3. // Returns the string s n times

```
// For example mult_string("hello", 3) would return
// "hellohellohello".
string mult_string(string s, int n) {
    string total;
    for (int i = 0; i < n; ++i)
        total = total + s;

    return total;
}
```

Complete program using functions

```
#include <iostream>
using namespace std;
// M is a global variable
int m;
int f(int n) {
    // Here a *copy* of the argument is modified, *not*
    // the object which was passed
    n = n + 1;
    return n;
}
int g() {
    // Here the global variable m is increased
    m = m + 1;
    return m;
}

int main() {
    // This n is local to this function, and *not* the
    // same thing as the n in the function f.
    int n = 5;
    cout << "n = " << n << "\n";
    cout << "f(n) = " << f(n) << "\n";
    cout << "n = " << n << "\n";

    // This is the global m declared at the beginning
    // of the program. It is the same m which is
    // modified in the function g.
    m = 5;
    cout << "m = " << m << "\n";
    cout << "g() = " << g() << "\n";
    cout << "m = " << m << "\n";
}
```

gives the output

```
n = 5
f(n) = 6
n = 5
m = 5
g() = 6
m = 6
```

Average of user defined numbers:

```
#include <iostream>
using namespace std;
// Print a line of '-' to divide one average from the
// other.
void delimiter() {
    for (int i = 0; i < 79; ++i)
        cout << "-";
    cout << "\n";
}
// Ask for n numbers and calculate their average.
void average(int n) {
    double sum = 0;
    for (int i = 1; i <= n; ++i) {
        // Ask for a number
        cout << "Enter number " << i << " of " << n
            << ": ";
        double num;
        cin >> num;
        // Add the number to the sum
        sum = sum + num;
    }
    // Print the average
    cout << "The average is: " << sum / n << ".\n";
}

int main() {
    // A for(;;) loop repeats for ever. We use 'return'
    // to jump out of the function directly
    for (;;) {
        cout << "How many numbers do you want to "
            <<"calculate the average of (0 to exit): ";
        int num;
        cin >> num;
        // Jump out of the function if the user entered zero
        if (num == 0)
            return 0;
        // Do the average and display a delimiter (line)
        average(num);
        delimiter();
    }
}
```

Use of Structures:

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

struct Data {
        string name;
        int age;
    };

int main() {
    Data person;
    person.name = "Sajid";
    person.age = 123;

    cout << person.name << " is " << person.age << " years old.\n";
}
```

Use of Classes

```
#include <iostream> // for cin and cout
#include <string>
using namespace std;
class Data {
public:
    void read() {
        cout << "Name: ";
        cin >> name;
        cout << "Age: ";
        cin >> age;
    }
    void print() const {
        cout << "Name: " << name << "\nAge: " << age
             << '\n';
    }
private:
    string name;
    int age;
};
```

```
void f(const Data& data) {
    Data.print();
}

int main() {
    Data some_data;
    some_data.read();
    f(some_data);
}
```