

# レポート第1回 「レポートの書き方 & C++プログラムの初歩」

氏名：大川 尚音  
科類・クラス：理科1類37組  
学生証番号：J4-231074  
E-mail: na-okawa04@g.ecc.u-tokyo.ac.jp

2024年10月8日

## 1 課題1 - 例1

### 1.1 プログラムリスト

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     cout << 4.0 * 3.14159265358979 * 6371000.0 * 6371000.0 << endl;
6 }
```

### 1.2 実行結果

```
プログラミング基礎 % g++ earth.cpp -o earth
プログラミング基礎 % ./earth
5.10064e+14
```

### 1.3 考察

この例に対しては、そもそも c++ の実行環境について考えていきたい。授業中に macOS 用に紹介されていた xcode について、apple 製品に対してはかなり最適なツールであったが、オーバースペックでまたローカル環境のストレージを圧迫していたので、今回は mac

に（おそらく）あらかじめ搭載されていた clang をラップした、g++を使用していくことにした。clang の version は 14.0.3 である。これまで主に扱っていた言語は python, typescript だったので、何気にコンパイルを行ったのは初めてだった。コンパイルを行うため、開発段階で手間が多くなっているように感じ、また手軽な修正しにくく感じるが、その分実行が早く行われるということで今後その違いについても留意しながら学習を続けたい。

## 2 課題 1 - 例 3

### 2.1 プログラムリスト

```
1 #include <iostream>
2 using namespace std;
3
4 int main( ){cout <<
5 4.0 * 3.14159265358979
6 * 6371000.0 * 6371000.0
7 << endl;}
```

### 2.2 実行結果

```
プログラミング基礎 % g++ earth_example3.cpp -o earth_example3
```

```
プログラミング基礎 % ./earth_example3
```

```
5.10064e+14
```

### 2.3 考察

先ほどのコードに比べて、半角スペースの削除、また不自然な改行を行ったが、実行結果は問題なく同じだった。python では、インデントを削除するとエラーが発生したので、それに比べて機械側でこの差異は認識されないとわかった。しかし、人間が開発する以上やはりインデントや改行は適切に行われている方が開発がしやすいので、適切なレイアウトを整えていきたい。また、先の例とこのコードをそれぞれコンパイルしたアセンブリコードは、同じ内容になっているのではないかと考察したが、うまく確かめる術もなかったなので仮説にとどまっている。

## 3 課題 1 - 例 4

### 3.1 プログラムリスト

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     cout << 4.0 * 3.14159265358979 * 6371000.0 * 6371000.0 << endl;
6     cout << "4.0 * 3.14159265358979 * 6371000.0 * 6371000.0" << endl;
7 }
```

### 3.2 実行結果

プログラミング基礎 % g++ earth\_example4.cpp -o earth\_example4

プログラミング基礎 % ./earth\_example4

5.10064e+14

4.0 \* 3.14159265358979 \* 6371000.0 \* 6371000.0

### 3.3 考察

今回は main 関数の中に二つの命令を書いたコードになっている。このコードをコンパイルしたアセンブリコードを実行すると、この順番で命令が実行されていることが少なくともこの例では確認できる。また、凡庸な言い方をすると見た目は似てダブルクォーテーションが入っただけの違いだが、cout で出力されるものはそれぞれ数値型と char 型になっていて、それはダブルクォーテーションによって区別されることがわかる。また、数字はデフォルトで文字ではなく数値として認識されることがわかる。

## 4 課題 1 - 例 5

### 4.1 プログラムリスト

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

```

3
4 int main(){
5     cout << 4.0 * 3.14159265358979 * 6371000.0 * 6371000.0 << endl;
6     cout << "4.0 * 3.14159265358979 * 6371000.0 * 6371000.0" << endl;
7     cout << endl << endl;
8     cout << 4.0 * 3.14159265358979 * 6371000.0 * 6371000.0;
9     cout << "4.0 * 3.1415";
10 }

```

## 4.2 実行結果

プログラミング基礎 % g++ earth\_example5.cpp -o earth\_example5

プログラミング基礎 % ./earth\_example5

5.10064e+14

4.0 \* 3.14159265358979 \* 6371000.0 \* 6371000.0

5.10064e+144.0 \* 3.1415%

## 4.3 考察

endl によって改行が行われていることがわかる。これまでのコードでは  
<< endl

によって改行が行われていたが、このコードの 8,9 行目にはそれがないため出力は連続なものになっている。また、endl のみを出力することも可能であり、それによって不自然な空白が出力されてしまっている。また、

\n

との違いについて調べてみると、書き込まれているものをバッファリングしているか否かの違いがあるらしく、この程度のコードなら問題ないが、念のためおまじないとして endl を使用していきたいと思う。