# 3. Program Set 2

1. Test out relational operators

#include <iostream>

int main() {

int x, y;

std::cout << "Enter x y: ";

if (!(std::cin >> x >> y)) return 0;

std::cout << "(x > y) = " << (x > y) << "\n";

std::cout << "(x < y) = " << (x < y) << "\n";

std::cout << "(x == y) = " << (x == y) << "\n";

std::cout << "(x != y) = " << (x != y) << "\n";

std::cout << "(x > 0 && y > 0) = " << (x > 0 && y > 0) << "\n";

std::cout << "!(x == 0) = " << !(x == 0) << "\n";

}

———————————————————————————————————————

1. Truthiness in single values

#include <iostream>

int main() {

int n;

std::cout << "Enter any integer: ";

std::cin >> n;

if (n)

std::cout << "This counts as TRUE (non-zero)." << std::endl;

else

std::cout << "This counts as FALSE (zero)." << std::endl;

return 0;

}

———————————————————————————————————————

1. Switch-case example

#include <iostream>

int main() {

int choice;

std::cout << "1) Tea 2) Coffee 3) Juice 9) Exit";

std::cout << std::endl << "Choice: ";

std::cin >> choice;

switch (choice) {

case 1:

std::cout << "Tea selected\n";

break;

case 2:

case 3:

std::cout << "Coffee or Juice!\n";

break;

case 9:

std::cout << "Goodbye!\n";

break;

default:

std::cout << "Unknown option\n";

}

return 0;

}

———————————————————————————————————————

1. while example

#include <iostream>

int main() {

int x = 0;

while (x < 5)

{

x = x + 1;

}

std::cout << "x is " << x << "\n";

}

———————————————————————————————————————

1. do-while example

#include <iostream>

int main() {

int n;

do {

std::cout << "Enter a number between 1 and 10: ";

std::cin >> n;

} while (n < 1 || n > 10);

std::cout << "You chose " << n << "\n";

return 0;

}

——————————————————————————————————————————————————————————————————————

1. for-loop example

#include <iostream>

int main() {

for (int i = 0; i < 10; i++)

{

std::cout << i << (i == 9 ? '\n' : ' ');

}

}

——————————————————————————————————————————————————————————————————————

1. Nested if example

#include <iostream>

int main() {

int x, y;

std::cout << "x y: ";

std::cin >> x >> y;

if (x > y)

{

std::cout << "x > y\n";

if (x == 6)

std::cout << "Also, x == 6\n";

else

std::cout << "Also, x != 6\n";

}

else

std::cout << "x <= y\n";

return 0;

}

——————————————————————————————————————————————————————————————————————

1. Rock-paper-scissors

#include <iostream>

#include <cstdlib>

#include <ctime>

int main() {

std::srand(static\_cast<unsigned>(std::time(nullptr)));

std::cout << "Enter your move (r/p/s): ";

char you;

std::cin >> you;

int comp = std::rand() % 3;

std::cout << std::rand() << std::endl;

char mv[3] = {'r','p','s'};

char c = mv[comp];

std::cout << "Computer: " << c << std::endl;

if (you == c)

std::cout << "Draw" << std::endl;

else if ((you=='r' && c=='s') || (you=='p' && c=='r') ||

(you=='s' && c=='p'))

std::cout << "You win!" << std::endl;

else

std::cout << "You lose!" << std::endl;

return 0;

}

——————————————————————————————————————————————————————————————————————

1. Number guess

#include <iostream>

#include <cstdlib>

#include <ctime>

int main() {

std::srand(static\_cast<unsigned>(std::time(nullptr)));

int target = 1 + std::rand() % 100;

std::cout << "Guess my number (1..100)" << std::endl;

int tries = 0, x;

for (std::cout << "Your guess: "; std::cin >> x; std::cout << "Your guess: ")

{

++tries;

if (x < target)

std::cout << "Too low!" << std::endl;

else if (x > target)

std::cout << "Too high!" << std::endl;

else

{

std::cout << "Correct in " << tries << " tries." << std::endl;

break;

}

}

}  
——————————————————————————————————————————————————————————————————————

1. Number base converter

#include <iostream>

#include <string>

int main()

{

unsigned long n;

int base;

std::cout << "Enter a non-negative integer and base (2, 8, or 16): ";

std::cin >> n >> base;

if (!(base == 2 || base == 8 || base == 16))

{

std::cout << "Base must be 2, 8, or 16.\n";

return 0;

}

if (n == 0)

{

std::cout << "0\n"; return 0;

}

std::string out;

while (n > 0)

{

int r = int(n % base);

char ch = r < 10 ? char('0' + r) : char('A' + (r - 10));

out += ch; // build the result in reverse

n /= base;

}

// print the reverse resuult which is itself in reverse

for (int i = int(out.size()) - 1; i >= 0; --i)

std::cout << out[i];

std::cout << "\n";

return 0;

}

——————————————————————————————————————————————————————————————————————