CSCE 120: Exam 1 Cheat Sheet

Here is an example of a simple C++ program:

```
//Looping Syntax
for (int i = 0; i < n; ++i) {
    if (some_condition) {
        continue; // Skip this iteration
    }
}


//cin && cout
int main() {
    int age;
    std::cout << "Enter your age: "; // Output
    std::cin >> age; // Input
    std::cout << "You are " << age << " years old!" << std::endl;
    return 0;
}


// common data types
int main() {
    int a = 5;         // Integer
    double b = 3.14;   // Floating point
    char c = 'A';      // Character
    bool d = true;     // Boolean

    return 0;
}


//if else blocks
int main() {
    int number = 10;

    if (number > 0) {
        std::cout << "Number is positive" << std::endl;
    } else {
        std::cout << "Number is not positive" << std::endl;
    }

    return 0;
}


//switch statements
int main() {
    int day = 3;
```

```
    switch (day) {
        case 1:
            std::cout << "Monday" << std::endl;
            break;
        case 2:
            std::cout << "Tuesday" << std::endl;
            break;
        default:
            std::cout << "Unknown day" << std::endl;
    }
    return 0;
}


//List syntax
#include <iostream>
#include <list>

int main() {
    // Create a list of integers
    std::list<int> numbers;

    // Add elements to the list
    numbers.push_back(10);  // Add to the end
    numbers.push_front(5);  // Add to the front

    // Iterate and print the list
    std::cout << "Elements in the list: ";
    for (std::list<int>::iterator it = numbers.begin(); it != numbers.end(); ++it) {
        std::cout << *it << " ";
    }
    std::cout << std::endl;

    // Remove elements from the list
    numbers.pop_back();   // Remove last element
    numbers.pop_front();  // Remove first element

    // Check if the list is empty
    if (numbers.empty()) {
        std::cout << "The list is now empty." << std::endl;
    }

    return 0;
}


//Lambda Functions
auto add = [](int a, int b) { return a + b; };
std::cout << add(2, 3); // Output: 5
```

```cpp
//std::setw for Formatting Output
#include <iostream>
#include <iomanip>

std::cout << std::setw(10) << "Name"
          << std::setw(5) << "Score" << std::endl;


//Handling Edge Cases with Functions
long long SumBetween(long long a, long long b) {
    // Handle edge cases for overflow
    if (a == LLONG_MIN || b == LLONG_MAX) {
        // Handle overflow
    }
    return a + b;
}


// Working with Parallel Arrays
std::vector<std::string> drivers = {"Driver A", "Driver B"};
std::vector<int> rankings = {1, 2};
for (size_t i = 0; i < drivers.size(); ++i) {
    std::cout << drivers[i] << ": " << rankings[i] << std::endl;
}


// Function to demonstrate pass by value
void passByValue(int x) {
    x = x + 10;  // This change won't affect the original variable
}
// Function to demonstrate pass by reference
void passByReference(int& x) {
    x = x + 10;  // This change will affect the original variable
}

int main() {
    int num = 5;
    passByValue(num);

    std::cout << "After pass by value, num = " << num << std::endl;
// Output: 5

    num = 5;
    passByReference(num);

    std::cout << "After pass by reference, num = " << num << std::endl;
// Output: 15
    return 0;
}
```

```cpp
// pointers
int main() {
    int arr[3] = {10, 20, 30};
    int* ptr = arr;  // Points to the first element

    for(int i = 0; i < 3; i++) {
        std::cout << *ptr << "␣";  // Dereference to get the value
        ptr++;  // Move to the next memory address
    }

    return 0;
}

// while/for loops
// 1d and 2d arrays
// if-else
// boolean conditions
// try/catch/throw
// cstrings
// structs
```