Welcome to OOP244

Introduction to OOP with C++

About me

- Dr. Nargis Khan
- Office: **jcz0uyp (Microsoft teams joining code)**
- nargis.khan@senecacollege.ca

Workshops and Final Project

- There will be 9 workshops and among them 8 will be counted. (30%)
- One final project divided in to 5/6 milestones (20%) –must submit

Workshops

- There will be 9 workshops. Some of the workshops have 1 part and some have 2 parts.
- Workshops 1 to5; two parts (part one guided, part two open ended)
- Workshops 6 to 9; one part (guided)
- Released every Friday
- Part 1 and workshops 6 to 9; due on Wednesday
- Part 2 Due Sunday
- **Deadlines are important!** The parts that are submitted after the deadline will receive 0 points. Incomplete workshops will receive 0 points.
- Code submitted late, get zero and rejected one day later.
- Best 8 workshops will be counted and from which 2 lowest “marks” will be dropped

Submission

- **Every** file that you submit as part of a workshop must contain the following information at the top:
- your name (as it is on BlackBoard)
- your student ID
- your Seneca email
- Your section name
- Use the following template:
- ***/***
  * Name:***
  **** Student ID:***
  **** Seneca email:***
  **** Section:***
  /

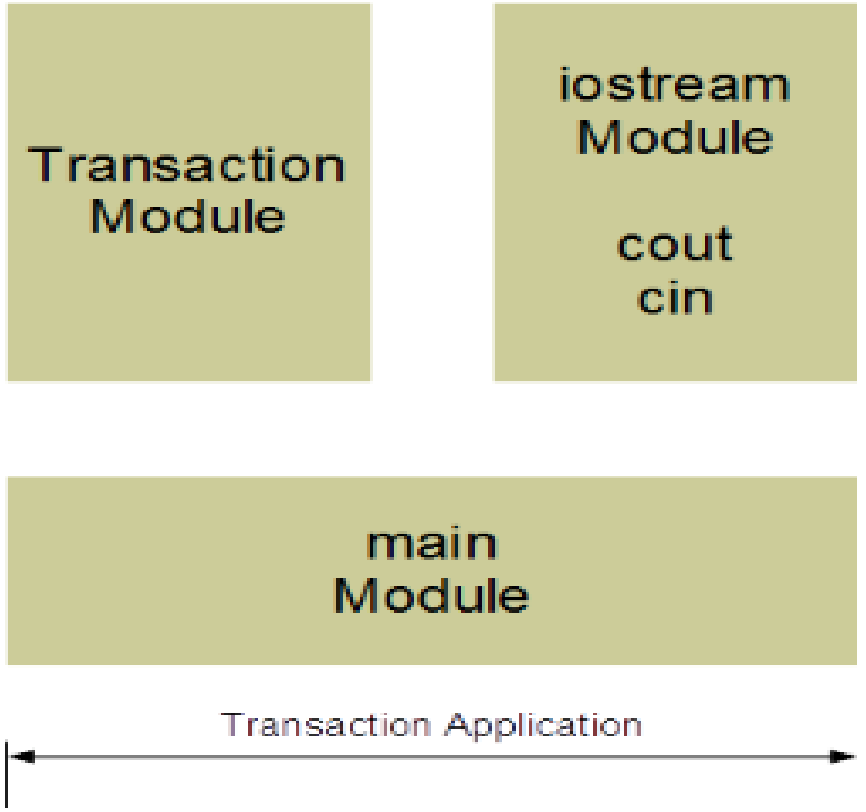
C program

```
#include <stdio.h>

int main ( )
{
printf("Welcome to c\n");
 }
```

Simple Program

```
// A Language for Complex Applications
// welcome.cpp
// To compile on linux:  g++ welcome.cpp
// To run compiled code: a.out
// To compile on windows:   cl welcome.cpp
//  To run compiled code: welcome


#include <iostream>
using namespace std;
int main() {
cout << "Welcome to Object-Oriented" << endl;
}
```

Transaction application



File extension

- .h or .hpp identifies the header file
- .cpp identifies the implementation file
- Standard C++ libraries do not include a file extension (<iostream>)

Transaction.h

- // code for Transaction.h
- Structure
- Function prototype

Transection.cpp

- Includes iostream
- Includes Transaction.h
- //code for transaction

main.h

```
//The header file for our Main module
// #defines the number of transactions:


#define NO_TRANSACTIONS 3
```

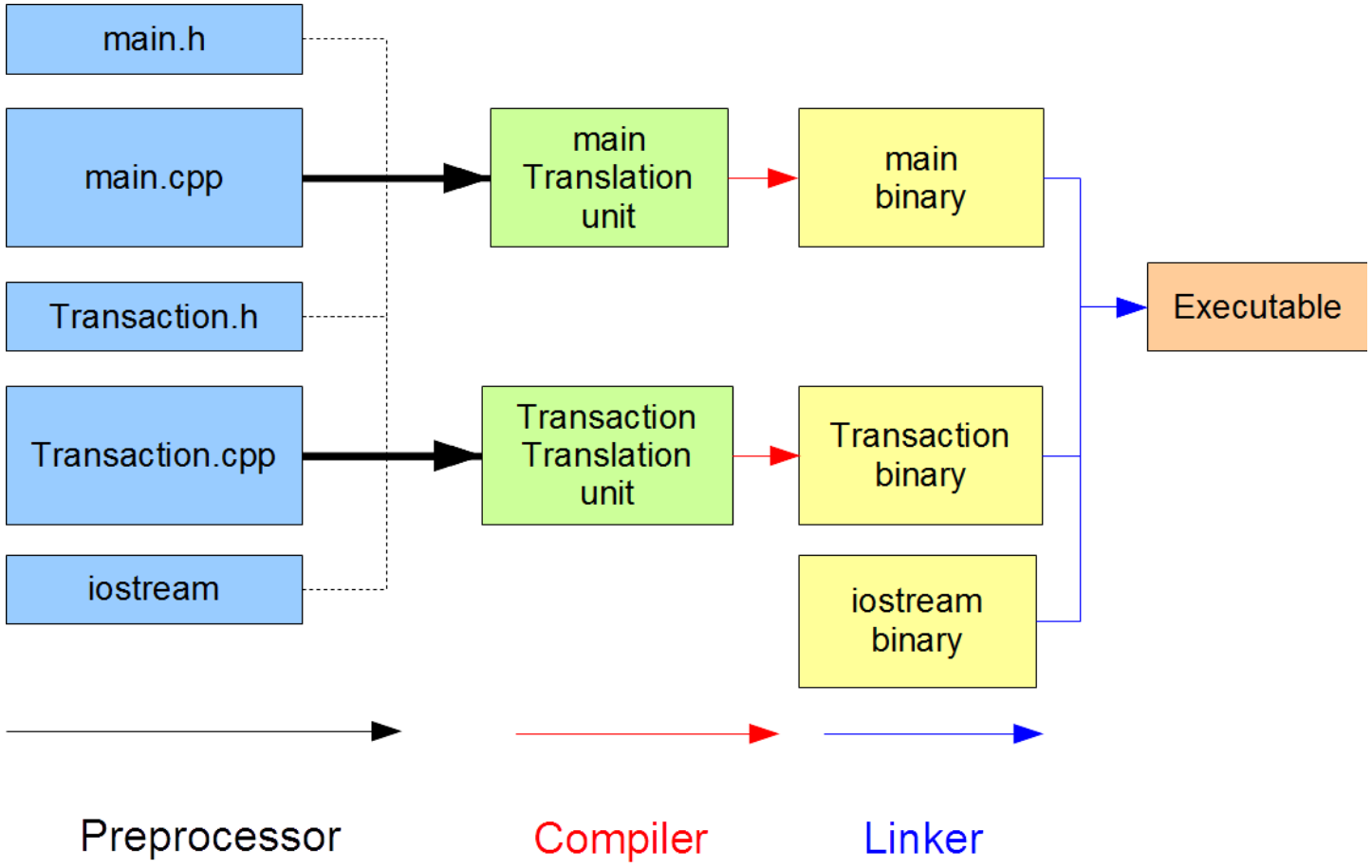
main.cpp

- Include main.h

- Include Transaction.h ~~Transaction.cpp~~
- (do not include)

- //code for main

STAGES OF COMPILATION



1.Preprocessor - interprets all directives creating a single translation unit for the compiler - (inserts the contents of all `#include` header files), (substitutes all`#define` macros)

2.Compiler - compiles each translation unit separately and creates a corresponding binary version

3.Linker - assembles the various binary units along with the system binaries to create one complete executable binary

COMPILE in command line

Open a developer command prompt:

1.Visual Studio 2018 on Windows 10, open the Start menu and choose **All apps**. Scroll down and open the **Visual Studio 2018** folder
Choose **Developer Command Prompt for VS2018** to open the command prompt window.
If you are using a different version of Visual Studio or are running a different version of Windows, look in your Start menu or Start page for a Visual Studio tools folder that contains a developer command prompt shortcut.

2. Next, verify that the Visual C++ developer command prompt is set up correctly. In the command prompt window, enter cl and verify that the output looks something like this:

Output

```
C:\Program Files (x86)\Microsoft Visual Studio 14.0>cl Microsoft (R) C/C++ Optimizing Compiler Version 19.00.23918 for x86
Copyright (C) Microsoft Corporation. All rights reserved. usage: cl [ option... ] filename... [ /link linkoption... ]
```

#

Compile in command line

Create a Visual C++ source file and compile it on the command line

1.In the developer command prompt window, enter **md c:\hello** to create a directory, and then enter **cd c:\hello** to change to that directory. This is the directory that your source file and the compiled program are created in.

2.Enter **notepad hello.cpp** in the command prompt window.

3.In Notepad, enter the following lines of code:
C++

```
include <iostream>
using namespace std;
void main() {

cout << “Welcome to object-oriented\n” << endl;
}
```

Compile in command line

- Simple program
  - cl hello.cpp
  - hello
- Moduler programming
  - cl \Fe anyname main.cpp transaction.cpp
  - anyname

Linux

To compile our application on a Linux platform at the command line, we enter the following

```
g++ -o accounting main.cpp Transaction.cpp
```

The `-o` option identifies the name of the executable. The names of the two implementation files complete the command.

To run the executable, we enter

```
accounting
```

Submission

- matrix.senecac.on.ca
- Upload your files in your matrix account
- Submission command:
- ~nargis.khan/submit 244_w1_lab
  - Where w1 will change according to lab number
- In lab in due in lab time (extension)
- At home due two days after lab day.
- DIY – within 5 days of the lab day

Submission

- All your work Must contain
- Your name
- Seneca email
- Student number
- Section
- Have to submit all parts to get marks other wise 0