Course Syllabus
CP470 Android Programming
Department of Physics and Computer Science/ Master of Computer Science
Faculty of Science, Waterloo Campus
Fall 2020

Professor Information

Dr. Abdul-Rahman Mawlood-Yunis | Science Building, N2078B
Email: amawloodyunisATwlu.ca | Phone: 519-884-0710 ex: 4091

Office Hours: Office Location: |By appointment or email|virtual/F2F

Course Information

As the worldwide smartphone market continues to grow, so does the demand for mobile applications. This course provides students with the skills for creating and deploying applications for mobile devices using Android, the most widely used operating system. With an emphasis on the Model-View-Controller paradigm. This course provides students with the foundational knowledge that underlies many popular programming languages. The course cumulates with the development of an original Android application.

Course name: CP470/Cp670 Android Application Programming, undergraduate/graduate Level, 0.5 credit

Requirements

Students are expected to possess good knowledge in Object-oriented programming and knowledge of Java is required. To take this course, students must own or have access to a windows computer that can run the latest Android Studio, Android Virtual Device (AVD) and Gradle build tool.

Course Overview and Approach/Framework

Android is a mobile operating system created for Android phones, Tablets, Android watches and TVs. Students will learn the application programming interface (API) of Android. Android's mobile apps are created by programming the API in a way that results in desired features. Android Studio is the integrated development environment (IDE) for writing Android apps. Android is free and runs on a windows computer. Students will learn how to efficiently use Android Studio. Mobile programming is a part of the larger discipline of application programming. In application programming, the programmer uses the API of a system to make applications for that system. A set of APIs is typically provided by an IDE. For example (besides our example on the Android), in Windows application programming, the programmer uses Eclipse to make programs (applications) running on a Windows computer. Application programming is the most fun part of computer programming. The importance of mobile apps in our everyday life is undeniable. Indeed, it is difficult to imagine living our lives well without a mobile phone and the apps in it. As a mobile application programmer, you can positively affect the lives of many people.

In this course, students will write a sequence of apps with increasing complexity. A new and concrete skill is learned in making the app. It is expected that by the end of the course, students would have acquired the skills needed to be a professional Android programmer. They will have to demonstrate their competence by making a commercial grade app.

Students are expected to spend on average 10 hours a week (reading, learning, coding) on this course. The instructor may be contacted by email or phone. If necessary, weekly online meetings can be arranged.

Course Goals and Learning Outcomes

This course aims to train students to be professional Android developers. By the end of this course students should be able to:

1. Understand fundamental concepts of Android mobile application programming and Java API Framework
2. Develop proficiency in Android programming, using Android Studio, Android Virtual Device (AVD), SDK manager, Graphical layout and Gradle
3. Develop proficiency in Android Application Programming interface (Android API)
4. Develop skills to write a graphical user interface for Android Mobile Phone using Model-View-Controller and event-driven techniques.
5. Develop Android Apps that support internationalization (multilingual support)
6. Handle app with multiple views
7. Develop proficiency in making games/Apps for the Android Phone
8. Develop proficiency in handling persistent data on the Android Phone
9. Develop proficiency in multi-thread programming and programming live videos and audio streaming on Android devices
10. Develop proficiency in testing and performance code evaluation

Course Tools and Learning Materials

There is no textbook for the course. Lecture notes will be provided at the course's website. There are several external websites, especially those maintained by Google, and Oracle for Java, that are useful:

1. Android guide and tutorial : https://developer.android.com/guide/
2. Android API https://developer.android.com/reference/
3. Android Studio https://developer.android.com/studio/
4. Add App Resources https://developer.android.com/studio/write/add-resources
5. Java tutorial https://docs.oracle.com/javase/tutorial/
6. Library - http://library.wlu.ca/
7. Learning Management System course login – http://mylearningspace.wlu.ca
8. Centre for Student Success (writing center, math center, academic advising, study skills/supplemental instruction, accessible learning) - http://www.wlu.ca/learningservices

Required Hardware and Software

- For your tests, you are going to use the Respondus Lockdown Browser and may use a built-in or an external webcam/phone.
- You are responsible for having a computer for doing your assignments and exams.
- "Final exams will be proctored virtually, with an external webcam or mobile phone.
- Zoom will be used for course communication, see this link (https://support.zoom.us/hc/enus/articles/201362033-Getting-Started-on-Windows-and-Mac to learn how to set it up.

Weekly Schedule(s)

Below is the weekly schedule proposed for the course. The order of topics to be covered may be changed in response to student progress.

Week 1	**JAVA review, Getting started with Android, First Android Application**
Week 2	**Debugging and Testing Android Applications**
Week 3	**Activity Lifecycle and Parcelable Interface**
Week 4	**User Interface**
Week 5	**ListView, Array Adapter, Date and Time Pickers and RecyclerView**
Week 6	**Reading week**
Week 7	**Toolbar, Menu, Dialog boxes, Shared Preferences and File Storage**
Week 8	**Android SQLite Database**
Week 9	**Parsing XML with XMLPullParser**
Week 10	**Fragments and Dynamic Layout Binding**
Week 11	**Services, Broadcast Receivers, and MediaPlayer**
Week 12	**Project Presentation**
Week 12	**Project Presentation**

Student Evaluation

Assignments (1 @ 2%, 2 @13% and 3@10)	25%
Quizzes (5 @2% each)	10%
Participation in online class discussions	5%
Project	30%
Research (group work)	5 %
Mid-term test	10%
Final Exam	15%
Total	**100%**

Summary of course deliverables and due dates:

CP 470/670 Forum (5%)

A critical piece of the online learning experience is to get to know your peers. I encourage you to connect with your classmates in the cp 470/670 forum. This is a graded discussion board where you can post your questions and share your knowledge by answering others' questions. You will be graded based on your contribution to creating new thread topics, replying to threads, and reading threads.

[Threads] [Replies] [Read (including own)]

To find the forum, click on the **discussions** tab on the toolbar above and select "470/670 Forum"

All three assignments (25%)

Assignment 1 (2%), No later than 11:59 p.m. on Day 7 of Week 2

Assignment 2 (13%), (delivered in four releases):
Release 1 and 2 for 6%, No Later than 11:59 pm on Day 7 Week 3
Release 3 for 3%, No Later than 11:59 pm on Day 7 Week 6
Release 4 for 4%, No Later than 11:59 pm on Day 7 Week 8

Assignment 3 (10%) (delivered in 3 releases)
Release 1 for 3%, No later than 11:59 p.m. on Day 7 of Week 10
Release 2 for 4%, No later than 11:59 p.m. on Day 7 of Week 11
Release 3 for 3%, No later than 11:59 p.m. on Day 7 of Week 12

Group research (Review paper) 5%

As a group, you will write an essay or a research paper of 5 to 10 pages on Introduction to Android or Android API evolvement. For example, you may write about:

Android OS or Android API evolvement, Java vs. Kotlin, Survey of Android IDE and Tools, a combination of these topics or a topic of your choice.

Group Project (30%) of five students:

Proposal	5%	**3rd week**
Iteration one code	10%	**7th week**
Presentation	5%	**11/12th** week (The project should be completed at this stage. You are allowed to make minor changes before uploading the final project to the mylearning space.)

Final Project deliverable (documentation, code, data, and testing) 80% **12th week**

Project Bonus Marks

2% bonus marking if you do your unit testing in **Junit**.

The app you will develop for your project must be a native mobile app; it must be developed using the Java programming language and Android Studio IDE. Projects developed using a third-party framework and then converted to an Android App will not be marked.

If there is a discrepancy between this syllabus and the lessons (or any course material), the syllabus will take precedence.

Learning Activities, Assignments, Tests, Quizzes and Examinations

Assignments (25%)

Assignments may be modified depending on the instructor's interests and the students' progress.

Grading Rubric for Assignments

Type of deductions	Examples	Amount of deductions

Major error/omission	• A required major feature is not implemented • Instructor must make one change to the code to compile the project	-7.5
Minor error/omission	• A minor feature is not implemented • Insufficient internal documentation	-2.5
Not compiled	The assignment cannot be compiled	-25 (no credit for assignments that do not compile)
Crash	The app crashes during operations	-12.5 (if some features are implemented) -25 (if no features can be seen due to the crash)
Every Assignment/Project must have its own app icons. Apps with no custom icons will be subjected to a 10-point deduction to total mark.		
The above schema is applied when marking your assignments; i.e., the deductions will be taken from your assignment mark. Assignments are graded out of 25 marks and converted to a course percentage worth up to 2% for Assignment 1, 13% for Assignments 2 and 10% for assignment 3.		

Assignments can be found in the "assessment description" section under the table of contents and will only be released during the week that they are due [e.g., you will be able to access Assignment 1 anytime (Day 1 to Day 7) during Week 2]. Once you have completed your assignment please submit it through the appropriate dropbox in MyLS. To locate the appropriate dropbox select "Dropbox" at the top of your MyLS webpage and then choose the "Assignment #" when you are ready to submit your assignment. See further information and policies below under "Assignment Policies".

Project (30%)

The group project requires that you utilize what you have learned in this course to develop an application for an Android device. The list of the minimum requirements for the project is as follows. You can be creative and go beyond the minimum requirements.

1. The app's main page must have multiple sections and 2 to 5 activities written for each section in the app. The activity must be accessible by selecting a graphical icon from a Toolbar and the main page section.
2. Each Section must use a **fragment** in its graphical interface.
3. Each Section must have a **ListView** to present items.
4. Selecting an item from the ListView must **show detailed information** about the item selected.
5. The items listed in the ListView must be **stored** by the application so that they appear the next time the application is launched.
6. The user must be able to **add** and **delete items**, which would then also be stored.
7. Each activity must use an **AsyncTask** in the code. This can be to open a Database, retrieve data from a server**, save data**, or any other reasonable circumstance.
8. Each Section must have at least 1 **progress bar**
9. Each Section must have at least 2-5 **buttons**
10. Each Section must have at least 1 **edit text** with an appropriate **text input method**.
11. Each Section must have at least 1 **Toast**, **Snackbar**, and **custom dialog notification**.
12. A help **menu item** that displays a dialog with the author's name, Activity version number, and instructions for how to use the interface.
13. There must be at least 1 other **language supported** by your Activity. If you are not bilingual, then you must support both British and American English (words like colour, color, neighbour, neighbor, etc). If you know a language other than English, then you can support that language in your application and don't need to support American English.
14. **Using content providers, firebase and animation are optional but encouraged. (is a plus)**

The project entails five parts:

1. Proposal (project charter)
2. App code
3. Project documentation
4. Testing
5. Student participation records

The five parts will be submitted as follows:

1. **Proposal:** Students can develop an app of their choice. However, students need the instructor's approval before starting to develop the app. When you submit your Project Charter or Description for approval, you need to include the following:

   - Problem Statement: Short and succinct (one or two sentences)
   - Project Objectives: What the project will achieve
   - Stakeholders: Persons who will be actively involved with the project (e.g. project sponsor, types of users, etc.)
   - Project Deliverables: The major results or services that will be produced, what are the specific things the software will do

   In order for your project to be approved, you need to show that:
   - You are addressing an important problem/application
   - The problem/application presents technical challenges, which you will successfully solve
   - Your solution employs high-quality software design
   - This system will work reliably, and be easy to understand and use

2. **Iteration 1: Your app code.** The code deliverable is done in two iterations (two steps). The first one needs to be delivered on Week 7 to give me a chance to see your progress and if needed to provide feedback and direction.

3. **Testing**
   Your app needs to be tested thoroughly and you need to document your test cases. Use Junit to do your tests.
4. **Final Project Deliverable should include everything that you have completed for your project including**
   Project Charter
   Project Documentation
   a. The Description of the class and method you created for the app.
   b. The class diagram to show the structure of your classes and the relationship between them
   c. Use case description and use-cage diagram.

   Iteration 2
   Complete app code (iteration 2).
5. Student participation records and peer review
   Group project will be adjusted according to the group members' contribution to the project.
   If all group members equally contribute to the project then all will have same mark.

Project	**Items**	**Percentage of total Project Mark which is 30**	**Due date**
	Proposal	5%	3rd week
	Iteration one code	10%	7th week
	Testing	5%	12th week
	Final Project deliverable	75%	12th week
	Student participation records and peer review		12th week

Your final submission should include your app with its own app icon. Make sure you test your app on an Android phone or tablet if your app is designed for a tablet. If your app requires a username/password, you must include them in your submission.

The project app will be marked based on:

- Complexity
- Functionality
- Originality, and
- Easy to use

Once you have completed your project deliverables please submit it through the appropriate dropbox in MyLS. To locate the appropriate dropbox select “Dropbox” at the top of your MyLS webpage and then choose the appropriate project deliverable when you are ready to submit your project work. See

further information and policies below under “Assignment Policies” and “University and Course Policies.

Assignment policy

- Assignments and projects must be submitted at mylearningspace.
- Assignments and projects are not accepted by email.
- Assignments will be posted at least one week before the due date.
- Programs should be written in the generally accepted style (For example, see the course note, or the book Code Complete: A Practical Handbook of Software Construction, Second Edition 2nd Edition, by Steve McConnell)
- Programs are marked on correctness and style, including internal documentation.
- Even though we do look at your code, the primary objective of the assignments is to implement the required functionality.
- Programs should be user-friendly and should not crash on bad input. Programs should warn user on bad input when this is feasible.
- Your assignment will be graded “fail” if it (i) does not compile, or (ii) has warnings, or (iii) crashes or hangs, or (iv) does not implement all requirements, or (v) does not have adequate internal documentation.
- The markers should be able to run your assignment without making any change to it.
- A later submission will override an earlier submission. Only the latest submissions are marked.

How to name the programs: suppose your Laurier email is *shoe3453@mylaurier.ca* and you are submitting assignment 2, then the program/Android project should be named *shoe3453_a2*, that is, the name of the Android project is *<author_email>_<assignment_number>* where *<author_email>* is your Laurier email without the @-part, and <assignment_number> is obvious. The project should be submitted in a zip file. For example, the assignment 1 submission of Shoemaker is named *shoe3453_a1.zip*. to submit this zip file. This naming convention facilitates the tasks of marking for the course markers and instructor. It also helps you in organizing your course work. Failure to follow the requirements will result in 20% mark reduction.

Frequently encountered problems with assignment submission

Problem: I completed my assignment, but I did not upload my program to mylearningspace by the deadline because my Internet connection was down (or, because ftp did not work, etc.)

Solution: Do not wait to the last hour to submit the assignment. If you are trying to submit the assignment from home, and your Internet goes down, that is your own problem. Try to submit it 3 hours before the deadline.

Problem: I submitted the wrong file, or my zip cannot be uncompressed and I don't know why.

Solution: You can always download your submission and verify that it contains the right files. This does not take more than three minutes. You may resubmit as many times as you like, the newly submitted file will replace the existing file in mylearningspace. Note that you have to submit a zip file. If you resubmit your assignment after the deadline, it will be considered late.

Problem: The project runs on my computer, but not on the marker’s computer.

Solution: The likely cause is you added a resource to the project folder using the option "Link to files". In this case only the link to the resource is added to the project. When the project is run on another computer, the resource cannot be found. So, when you add a resource, choose the option "Copy files".

Problem: My assignment is strikingly similar to that of another student because we "worked together" on it.

Solution: The assignments are individual. ***Do not work with another student on them. Do not give your work to another student.*** *If you are charged with plagiarism and it is your first offense, your submitted work will receive a mark of 0, ten final marks will be deducted from you final mark total, a letter will go into your permanent record, and copies will be sent to the Chair of Physics & Computer Science, and the Dean of Faculty of Science. If it is your second offense, then … you should not even think about it. Since this is a serious matter, and plagiarism occurs frequently, we will make clear of this course's policy (modeled after that of several North American universities, in particular, Stanford University's policy):*

It is usually appropriate to ask others—the TA, the instructor, or other students—for hints and debugging help or to talk generally about problem-solving strategies and program structure.
The important point, however, is embodied in the following rule:

Rule 1: You must indicate on your submission any assistance you received.
If you make use of such assistance without giving proper credit, you may be guilty of plagiarism.
In addition to providing proper citation—usually as part of the comments at the beginning of the program—it is also important to make sure that the assistance you receive consists of general advice that does not cross the boundary into having someone else write the actual code. ***If you used code from the Internet, say it in your submission and provide the link to the code****. It is fine to discuss ideas and strategies with others, but you should be careful to write your programs on your own. This provision is expressed in the following rule:*

Rule 2: You must not share actual program code with other students.
In particular, you should not ask anyone to give you a copy of their code or, conversely, give your code to another student who asks you for it. Similarly, you should not discuss your algorithmic strategies to such an extent that you and your collaborators end up turning in exactly the same code. Discuss ideas together, but do the coding on your own. The prohibition against looking at the actual code for a program has an important specific application in computer science courses. Developing a good programming assignment often takes years. When a new assignment is created, it invariably has problems that require a certain amount of polishing. To make sure that the assignments are as good as they can be, this department—like most others—reuses assignments over the years, incorporating a few changes each time to make them more effective. The following rule applies:

Rule 3: You must not look at solution sets or program code from other years.
Beyond being a clear violation of academic integrity, making use of old solution sets is a dangerous practice. Most assignments change in a variety of ways from year to year as we seek to make them better. Each year, however, some student turns in a solution to an assignment from some prior year, even though that assignment has since changed so that the old solution no longer makes sense. Submitting a program that solves last year's assignment perfectly while failing to solve the current one is particularly damaging evidence of an academic integrity violation. Whenever you seek help on an assignment, your goal should be improving your level of understanding and not simply getting your program to work. Suppose, for example, that someone responds to your request for help by showing you a couple of lines of code that do the job. Don't fall into the trap of thinking about that code as if it were a magical incantation— something you simply include in your program and don't have to understand. By doing so, you will be in no position to solve similar problems on exams. The need to understand the assistance you receive can be expressed in the following rule:

Rule 4: You must be prepared to explain any program code you submit at any time.
We may perform the following procedure to detect academic violations. We may use plagiarism detection tools. We archive all submissions, both from this semester and previous semesters, and cross-compare for unusual resemblance. We do not target specific students, all assignments are subject to the same scrutiny. Any similarity detected by the tools is then examined more closely by the course's staff and appropriate actions will be taken. The tools are very adept at identifying all variants of improper collaboration, from major to minor.

Rule 5: All submissions are subject to automated plagiarism detection.

In summary
Although you should certainly keep these rules in mind, it is important to recognize that the cases that we bring forward to the Dean's office are not those in which a student simply forgets to cite a source of legitimate aid. Most of the students we charge have committed fairly egregious violations. Students, for example, have rummaged through paper recycling bins or undeleted trash folders to come up with copies of other students' programs, which they then turn in as their own work. In many cases, students take deliberate measures— rewriting comments, changing variable names, and so forth—to disguise the fact that their work is copied from someone else. Despite these cosmetic changes, it is usually easy to determine that a copy has been made. Programming style is highly idiosyncratic, and the chance that two submissions would be that similar is vanishingly small. We have no desire to create a climate in which students feel as if they are under suspicion. The point is that we all benefit from working in an atmosphere of mutual trust. Students who deliberately take advantage of that trust, however, poison that atmosphere for everyone. As members of the Laurier community, we have a responsibility to protect academic integrity for the benefit of the community as a whole.

Quizzes (10%)
There will be five quizzes in the course each worth 2%. Your final quiz submission will be used to calculate your mark which will then be converted to a course percentage worth up to 2%. The purpose of these quizzes is for you to review and consolidate the new knowledge that you have gained. Quizzes will be available during the weeks they are due and accessed by selecting "Quizzes" at the top of your MyLS page and then selecting the appropriate quiz. Below you will find a chart that outlines what Lessons/Topics each quiz covers along with its percentage weight and the week that it is available/due.

Quiz	Lessons and Topics	Weight	Due date: no later than 11:59 p.m. on the last day of:
1	• Lesson 1, 2, 3, 4	2	4th week
2	• Lesson 5, 6	2	6th week
3	• Lesson 7, 8, 9	2	9th week
4	• Lesson 7, 8, 9, 10	2	10th week
5	• Lesson 9, 10, 11	2	11th week

Midterm, **10%**
The Midterm test will be available starting at 8:00 AM on Day 7 of Week 7
The midterm will test your understanding of material covered in **Lessons 1 through 6**. The test will include short concept questions. Some questions may include writing code. Some questions may be multiple-choice. The midterm will be conducted online using Respondus Lock down browser. The test

will be available starting at 8:00 AM on Day 7 of Week 7 and will close at 11:59 p.m. on the same day. Check the course calendar for an exact date. You will have 2 hours to write the online test. Although you can decide when to start your two 333hours, please note that once started you will not be able to pause or restart your test. To prepare for the test you should read the lessons, do the activities and exercises. The test is closed book. No material (the course lessons, your own notes, a computer, etc.) is allowed. When you are ready, you may access the midterm by clicking on "Quizzes" and then selecting "Midterm" at the top of your MyLS webpage.

Final 15%

The final exam is cumulative and will cover material from Lessons 1 through 13 but with more emphasis on Lessons 7-13. The Final exam will be scheduled by the Final Exams Office. Ensure you check with the Final Exam schedule to learn when the final exam for this course will be scheduled. To prepare for the exam you should read the lessons, do the activities and exercises. The exam is closed book. No material (the course lessons, your own notes, a computer, etc.) is allowed. The exam's format is the same as that of the midterm test. You will have 2 hours to write the online exam. Respondus Lock down browser will be used. You may access the final exam during the exam period by click on "Quizzes" and selecting "Final Exam".

To pass the course, students must obtain at least 50% of the assignments, %50 of the project, and 50% of the mid-term test and final exam (in combination, not separately).

University and Course Policies

1. **Academic Integrity/Misconduct** (cheating)**:** Laurier is committed to a culture of integrity within and beyond the classroom. This culture values trustworthiness (i.e., honesty, integrity, reliability), fairness, caring, respect, responsibility and citizenship. Together, we have a shared responsibility to uphold this culture in our academic and nonacademic behaviour. The University has a defined policy with respect to academic misconduct. You are responsible for familiarizing yourself with this policy and the penalty guidelines and are cautioned that in addition to failure in a course, a student may be suspended or expelled from the University for academic misconduct and the offense may appear on their transcript. The relevant policy can be found at Laurier's academic integrity website along with resources to educate and support you in upholding a culture of integrity. **Ignorance of Laurier's academic misconduct policy is not a defense. <**see: www.wlu.ca/academicintegrity **>**

2. **Special Needs:** Students with disabilities or special needs are advised to contact Laurier's Accessible Learning Centre for information regarding its services and resources. Students are encouraged to review the Academic Calendar <see: http://www.wlu.ca/page.php?grp_id=1365&p=5123 > for information regarding all services available on campus.

3. **Plagiarism:** Wilfrid Laurier University uses software that can check for plagiarism. If requested to do so by the instructor, students may be required to submit their written work in electronic form and have it checked for plagiarism. (Approved by Senate May 14, 2002)

4. **Communication Policy**
   Communication between your instructor and the class as a whole will be through the Newsfeed on MyLearningSpace (MyLS). Individual communication will be through MyLS e-mail. E-mails are responded to within 24hrs hours or less, except on weekends or holidays. Please check for an

answer to your question in the course syllabus and course roadmap before sending an e-mail, as many times the answer to most questions are found in one of these documents.

5. **Late Assignment Policy:** Students are expected to submit all assignments to the course professor on or before the due date in the format specified by the professor and as instructed. No make-up for missed assignments, tests, or exams will be permitted (except under exceptional circumstances as outlined in Laurier University policy). Any evaluation with a course grade weight up to and including 10% will not be accepted if submitted late, and a grade of F will be assigned (except under exceptional circumstances as outlined in Laurier University policy). Any assignment with a course grade weight of 11% or greater, if submitted late, will be penalized 20% per day, including weekend days, up to a maximum of four (4) days. Those assignments received after the fourth (4th) day will receive an F grade (except under exceptional circumstances as outlined in Laurier University policy).

6. **Final Examinations** – Students are strongly urged not to make any commitments (i.e., vacation) during the examination period. Students are required to be available for examinations during the examination periods of all terms in which they register. (See Academic Regulations – examinations in the academic calendars)

7. **Foot Patrol, The Wellness Centre, and the Student Food Bank** (Approved by Senate November 28, 2011 – see below)

Waterloo

Student Food Bank | www.wlusu.com/food-bank/ All students are eligible to use this service to ensure they're eating healthy when overwhelmed, stressed or financially strained. Anonymously request a package online 24-7. All dietary restrictions accommodated.
Foot Patrol – 519.886.FOOT (3668) A volunteer operated safe-walk program, available Fall and Winter, daily from 6:30pm to 3am. Teams of two are assigned to escort students to and from campus by foot or by van. http://www.wlusu.com/foot-patrol/
Peer Connect – 1.866.281.PEER (7337) A confidential listening, referral, information and support line, is available during evening hours to provide support and resources. Sunday to Thursday, 12pm – 2am | Friday to Saturday 12pm – 3am http://www.wlusu.com/peer-help-line/
The Wellness Centre | 519-884-0710, x3146 The Wellness Centre supports students' physical, emotional and mental health needs. Located on the 2nd floor of the Student Services building, booked and same-day appointments are available Monday to Wednesday 8:30 am – 7:30 pm, Thursday to Friday 8:30 am-4:15 pm. Contact: x3146, wellness@wlu.ca or @LaurierWellness. After hours crisis support available 24/7 - "Good 2 Talk" 1-866-925-5454.

Brantford

Student Food Bank | www.wlusu.com/food-bank/ All students are eligible to use this service to ensure they're eating healthy when overwhelmed, stressed or financially strained. Anonymously request a package online 24-7. All dietary restrictions accommodated.

Foot Patrol l 519-751-PTRL (7875) A volunteer operated safe walk program, available Fall and Winter, Monday-Thursday 6:30pm-1am and Friday-Sunday 6:30-11pm. Radio dispatched teams are available upon call to escort students to and from campus as well as off-campus destinations either by foot or by van. http://www.wlusu.com/foot-patrol/
Peer Connect – 1.866.281.PEER (7337) A confidential listening, referral, information and support line, is available during evening hours to provide support and resources. Sunday to Thursday, 12pm – 2am | Friday-Saturday 12pm – 3am http://www.wlusu.com/peer-help-line/
The Wellness Centre | 519-756-8228, x5803 Students have access to support for all their health and counselling needs at the Wellness Centre. Located in the Student Centre, 2nd floor. Hours: 8:30am to 4:30pm Monday-Friday. After hours crisis support available 24/7 - "Good 2 Talk" 1-866-925-5454.