

****Disclaimer****

This syllabus is to be used as a guideline only. The information provided is a summary of topics to be covered in the class. Information contained in this document such as assignments, grading scales, due dates, office hours, required books and materials may be from a previous semester and are subject to change. Please refer to your instructor for the most recent version of the syllabus.

CSE100 Principles of Programming with C++

Disclaimer: this web page is the official syllabus for the course; paper copies will not be distributed. During the semester, changes may be made to the syllabus. If the change is significant, notification will be given in class, and an announcement will be made in Blackboard. Minor editing changes will not be announced. The student is responsible for reading this syllabus at the beginning of the semester to acquaint himself or herself with the course policies, and for checking the syllabus periodically throughout the semester for relevant information.

all 2014 --- [Course Information](#)

Catalog Course Description

Lecture: Principles of problem solving using C++, algorithm design, structured programming, fundamental algorithms and techniques, and computer systems concepts. Social and ethical responsibility. Three (3) credit hours. Lecture, with required lab. Prerequisite: None.

Lab: This is a required fifty minute lab for students enrolled in the CSE100 Principles of Programming with C++ lecture. The purpose of the lab is for the student to practice writing computer programs using C++ with the TA present and available for help. It is taught in a computer lab, so each student will be able to work on his/her programs during the lab session.

Expanded Course Description

This class used to be the first required Computer Science & Engineering course for students pursuing the B.S. degree in Computer Systems Engineering (CSE). However, the School of CIDSE now recommends that both CS and CSE majors complete CSE110 Principles of Programming with Java. The primary population in CSE100 is now drawn from mathematics, physics, and various engineering disciplines. I understand that most of you are taking this course to fulfill a major requirement, and I teach it at a level that is appropriate for non-CS majors. You should understand that even though you may never become a programmer, or may never write another program again in your life, the experts in your field believe it is important for a student in that field to know something about computer programming. For many of you, you may very well end up writing quite a bit of software during your career, and this course will be immensely important to your education.

This is not necessarily a course on the C++ language, but rather, it is a course on computer programming principles and concepts. In particular, we focus on imperative/procedural and object-oriented programming principles. There are many programming languages which *could* be used to teach these concepts, including Java, Python, C#, and Visual Basic. Any one of these languages would suffice, but C++ has been selected because of its popularity and widespread use in both academia and industry —especially in engineering and the sciences.

At all times in this course, you are encouraged to focus more on the programming concepts being taught, and less on the specific C++ syntax. For example, we will learn about selection statements (also called decision statements or if statements). In C++, selection statements are implemented using a statement which begins with the specific word "if" and so they are commonly called "if statements". However, every programming language has some form of a selection statement. It may not be written in the same way as it is in C++, but trust me, it's there. So when we learn about selection statements, do not get so caught up in the details of the C++ "if" statement that you forget to learn the more general concept of what a selection statement is, how it works, and when and where you would use one.

Course Objectives

1. Develop programs that use fundamental data types, strings, arrays, and objects.
2. Use predefined library functions in programs.
3. Write programs with multiple user-defined functions.
4. Use flow of control structures.
5. Understand the pass-by-value and pass-by-reference parameter passing techniques.
6. Understand variable scope and use scope appropriately in programs.
7. Understand 1-D and 2-D arrays and use them appropriately in programs.
8. Develop object-oriented C++ programs using classes, information hiding, and inheritance.
9. Use the Unified Modeling Language (UML) to design and document classes.
10. Understand searching and basic sorting algorithms.

Major Topics

The material to be covered in this course will be drawn from Chapters 1-9 of the textbook, although not necessarily in the order listed here. Furthermore, we will discuss some material that is not in the text.

Chapter 1: Introduction to Computers and Programming

- Introduction to computer hardware.
- Introduction to software.
- Programming and programming languages.
- Programming tools.

Chapter 2: Introduction to C++

- Parts of a C++ program.
- #include preprocessor directive.
- Variables.
- Literals.
- Assignment operator, assignment statement.
- Identifiers.
- cin and cout.
- Integer and floating point data types.
- Variable scope.
- Arithmetic operators.
- Comments.
- Programming style.

Chapter 3: Expressions and Interactivity

- cin.
- Mathematical expressions, math library functions.
- Implicit and explicit type conversions.
- Named constants.
- Combined assignment operators.
- Input and output with characters and strings.
- Debugging.

Chapter 4: Making Decisions

- Relational operators.
- Logical operators.
- If statement, if-else statement, if-elseif-... statement.
- Testing and debugging.

Chapter 5: Looping

- While loop.
- Increment and decrement operators.
- Counter variables, counting loops.
- For loop.
- Sentinels and sentinel loops.
- Nested loops.
- Text file I/O.

Chapter 6: Functions

- Modular programming.
- Defining and calling functions.
- Function prototypes.
- Pass by-value parameters.
- Returning values from a function.
- References and pass by-reference parameter passing.

Chapter 7: Introduction to Classes and Objects

- Object oriented programming.
- Objects.
- Classes.
- Member functions.
- Constructors.
- Passing objects as parameters by-value and by-reference.
- Object composition.

Chapter 8: Arrays

- Array basics.
- Accessing array elements.
- Input and output with arrays.
- Array initializers.
- Processing arrays.
- Passing arrays as function parameters.
- Two dimensional arrays.
- Arrays of objects.

Chapter 9: Searching, Sorting, and Algorithm Analysis

- Algorithms.
- Linear search.
- Binary search.
- Selection sort.
- Bubble sort.
- Time complexity of searching and sorting algorithms.

Office Hours & Support

The instructor will be available during his **office hours** to answer any questions you may have about the course, the material, or the projects. I'm a friendly guy (except when I'm not), and willing to talk with you and help you as much as I can, so stop by if you want, but please be aware that I also teach other courses and have other duties as well, so I am quite busy. When you come in for help, please be prepared to ask pointed and specific questions about what you do not understand or are having difficulty with. It is next to impossible for me to help you if you

come in and say, "I don't understand anything." Also, a final word of advice. I may not be available as much as you would like the day before or the day an assignment **is due**. I will give you plenty of time to complete the assignments as long as you start working on them well before the deadline. This will give you ample time to meet with me, a graduate TA, an UGTA, or a grader to discuss your questions as you are working on the assignment. Start early!

The labs are taught by a graduate TA who will also grade some of the lab projects (most will be graded by the grading assistants, discussed below). The graduate TA is Subhasish Das (Subhasish[.dot]Das[@-sign]asu[.dot]edu). Subhasish will hold two office hours each week Mon 11:00–12:00 and Tue 12:00–1:00 in CTRPT 114.

There will be two grading assistants for the course. Each grader will conduct one office hour per week so you can meet with him or her to ask questions about the grading of a homework assignment, lab project, or exam (note: it would be best if you meet with the grader who graded your assessment).

We also have three UGTA's assigned to assist the graduate TA in teaching the labs. The UGTA's are: Raymond Barakat (Mon, Thu), Brandon Kral (Tue, Thu), and Peter Johnson (Tue, Thu). Unfortunately, we don't have a UGTA for the Wed lab section.

The School of Computing, Informatics, and Decision Systems Engineering (CIDSE) provides free tutoring in the **CIDSE Tutoring Center**, which is located in **CTRPT 114**, Mon through Fri 9:00am to 6:00pm.

The University Academic Success Programs will offer **small-group online tutoring**. Here is the **schedule for the first week of class**. The full schedule of tutoring will begin Tue 2 Sep and the schedule will be available on the online tutoring website beginning Fri 29 Aug. You will see another Blackboard course shell specifically for online tutoring. [Course Materials & Resources](#)

Course Materials & Resources

- **Pearson My Programming Lab (MPL)** – Combines the **online textbook** with **CodeLab**, an online programming and problem-solving environment by Turings Craft. You will register and pay the MPL fee (it is supposed to be \$91.30) on the **Pearson MPL website**. You will need an access code—available in Blackboard — for registration. Make sure to read **this note** regarding your ASU identification number.
- **My ASU Courses (Blackboard)**
- See the **Tools** section below.

The website you are currently viewing is the main website for this course (I will refer to it as the course website), and most course information will be available here (e.g., reading assignments, lab project documents, exam study guides, etc). I use Blackboard for: (1) posting important announcements; (2) student submission of lab project files; (3) posting exam solution and grading rubrics; and (4) posting scores in the Blackboard Grade Center. [ment](#)

Assessment

Various methods will be used to present the material and assess the student's understanding and comprehension.

Pearson MPL Exercises

Pearson MPL exercises for each chapter of the text will be assigned with deadlines for completion. MPL exercises will be available for 48 hours after the deadline, during which time you may still submit answers for a 20% penalty. During the course there will be a maximum of n MPL exercise points to be earned. At the end of the term, an MPL percentage, denoted mpl in the course percentage formula below, will be calculated as a percentage of the exercise points that were answered correctly divided by $0.90n$, not to exceed 100%, i.e., I will drop 10% of the MPL exercise points and your MPL percentage cannot exceed 100%.

Note: When you register for your MPL account, you must enter your 10-digit ASU student identification into your account information. Failure to do so will result in no points being awarded for the MPL exercises. To enter your identification number, after you log in, click on the gear icon in the upper right corner of the window. Click on Settings. In the Account Info area, enter your identification number in the text field labeled ID.

Note: in cases of academic integrity violations, and the assignment of a score of 0 on any MPL assignment, your MPL course percentage will be calculated as the sum of all earned MPL exercise scores divided by the number of MPL exercise points that were possible to be earned, i.e., no exercises are dropped.

Lab Projects

During each required lab session (12 lab sessions), you are required to complete a programming project which will be submitted to Blackboard for grading. Each project is worth five (5) points, but your lowest lab project score will be dropped, so there will be a maximum of 55 project points to be earned. Before each lab session, there may be assigned prelab exercises which you must complete before the lab session. Failure to complete the prelab exercises may make it very difficult for you to fully participate in the lab and complete the in-lab programming project. In general, lab projects will be due at 4:00am on Saturday mornings, approximately one week after the Friday lab sessions. Specific dates are found in the monthly calendars in the [Schedule](#) section. Lab projects are accepted for grading on a variable deadline (except for the final lab project which will be due the day of the final exam).

Submitted

Bonus/Penalty

> 48 hrs before the deadline	Bonus of 20% of earned pts.
> 24 hrs and $\leq$ 48 hrs before the deadline	Bonus of 10% of earned pts.
$\leq$ 24 hrs before the deadline	Accepted for grading with no bonus/no penalty.
< 24 hrs after the deadline	Penalty of 10% of assignment pts.
$\geq$ 24 hrs and < 48 hrs after the deadline	Penalty of 20% of assignment pts.
$\geq$ 48 hrs after the deadline	Not accepted for grading. Score will be 0 pts.

Projects that are submitted more than 24 hours early will be awarded bonus points. For example, if the project is worth 5 points and you earn 3.5 points by submitting the project 50 hours before the deadline, then you will be awarded $3.5 \times 20\% = 0.7$ bonus points, for a total project score of 4.2 points. Projects that are submitted less than 48 hrs after the deadline will be penalized points. For example, if the project is worth 5 points, and you earn 3.5 points by submitting your project one minute after the deadline, then you will be penalized $5 \times 10\% = 0.5$ points, for a total project score of 3 points. The penalty for 24 to 48 hours after the deadline is 20% of the project points. In no case will a project be graded when submitted 48 hours or more after the deadline. Project solutions must be uploaded to Blackboard for grading; in no case will an emailed project be accepted for grading.

At the end of the term, a lab percentage, denoted *lab* in the course percentage formula below, will be calculated as the sum of the points you earned on the 11 highest-scoring lab projects divided by 55. Your lab percentage, including bonus points, will be limited to a maximum of 100%.

Note: in cases of academic integrity violations, and the assignment of a score of 0 on any lab project, no lab project scores will be dropped, and the lab project percentage will be calculated as the sum of all earned lab project scores divided by the number of the lab project points that were possible on all 12 lab projects.

Examinations

There will be three in-class midterm examinations; the exam dates are listed on the monthly calendars in the [Schedule](#) section. There will be a noncomprehensive final exam given during the final exam period, see the [Dec calendar](#). The final exam will be not be comprehensive. Your lowest exam score, including the final, will be dropped when computing your final course percentage using the first formula, *CP1*. If you wish to base your grade solely on the exams, when calculating *CP2* no exam scores are dropped.

Only in exceptional cases (exceptional is defined by the instructor and will generally involve something like a note from the neurosurgeon who performed your emergency hemispherectomy on the day of the exam) will an exam be given after the exam date. If you know you are going to be absent on the date of an exam for a good reason (generally only business travel for your job or travel for an approved school function) then you may be permitted to take the exam early.

Note: you will be required to show proper photo identification (a driver's license, military ID card, or ASU Suncard) when handing in your examination for grading. If the instructor or TA does not recognize you, and you cannot produce photo identification, your exam will not be accepted for grading. Note: in cases of academic integrity violations, and the assignment of a score of 0 on any midterm exam or final exam, the 0 score will not be considered the lowest exam score and will not be dropped.

Quizzes

From time to time, there may be unannounced open-book extra-credit pop quizzes. In no case—never—will a makeup quiz be given if you miss class for any reason. A quiz percentage, denoted q in the course percentage formulas below, will be calculated as the sum of your earned quiz points divided by the sum of the quiz points on all of the quizzes and then multiplied by 2.5. The quizzes will be worth up to 2.5% extra credit which will be added to your final course percentage before assigning letter grades. Note that quizzes are essentially free attendance points.

Calculating Final Letter Grades

Your final letter grade will be based on your final course percentage FCP which is calculated as a weighted sum of your scores on the MPL exercises, lab projects, and examinations. Two course percentages will be calculated using these formulas:

$$CP1 = (mpl \times 5\%) + (lab \times 30\%) + (e1 \times 21.67\%) + (e2 \times 21.67\%) + (e3 \times 21.67\%) + q$$

$$CP2 = (mid1 \times 25\%) + (mid2 \times 25\%) + (mid3 \times 25\%) + (final \times 25\%) + q$$

Where mpl is your Pearson MPL percentage as described in the [Pearson MPL Exercises](#) section above; lab is your lab project percentage calculated as described in the [Lab Projects](#) section above; and q is your extra-credit quiz percentage as described in the [Quizzes](#) section above. In $CP1$ $e1$ is your highest exam score (midterm or final), $e2$ is your second highest exam score (midterm or final), and $e3$ is your third highest exam score (midterm or final). In $CP2$ all three midterm exam scores and your final exam score are used. The final course percentage will be calculated as:

$$FCP = \text{ceiling}(\max(CP1, CP2))$$

The *ceiling* of x is the smallest integer that is greater than x . For example, the ceiling of 79.2 is 80 and the ceiling of 89.99999 is 90. Your final letter grade will be based on FCP :

FCP	Letter Grade
$FCP \in [98, 100]$	A+
$FCP \in [85, 98)$	A
$FCP \in [70, 85)$	B
$FCP \in [55, 70)$	C
$FCP \in [40, 55)$	D
$FCP < 40$	E

Grade Appeals

If you believe a graded assessment (e.g., lab, quiz, or exam) was marked incorrectly, the first step is to meet with the TA or grading assistant who graded the assessment in question and discuss your concerns with him or her. Quizzes, exams, and some of the lab projects are graded by the grading assistants. The remaining labs are graded by your TA. If the TA or grading assistant made an honest mistake, then he or she should correct it and enter the updated score in Blackboard. On the other hand, if the TA or grading assistant correctly marked, or believe they correctly marked, the assessment using the grading rubric the instructor gave them, and after conferring with them, you disagree, then you may file a grade appeal with the instructor using [this form](#). Please read the form instructions carefully. Basically, you have one week from the date the lab, quiz, or exam was graded to file your grade appeal. No appeals will be accepted after the one week period for any reason. You must submit the original lab, quiz, or exam, along with your grade appeal form in person to the instructor; emailed forms will be discarded. Grade appeals are considered by the instructor at the end of the semester before calculating your final course percentage and letter grade (the reason is that in most cases, whether the appeal is successful or not, it will not

affect a student's course percentage enough to change the student's letter grade). At that time, I will regrade your assessment only if giving you the points in question would change your final course percentage enough to increase your final letter grade. If your score was changed, the new score will be entered in the Blackboard Grade Center.

Academic Misconduct

In general, the instructor believes learning is a collaborative activity, that students learn best when they work together, and that students should be encouraged to learn from and teach each other. These activities would include discussing solutions to MPL exercises, working together on lab projects, and jointly studying for exams. In completing the lab projects, student-pair collaboration is encouraged and will be permitted as long as each member of the pair contributes equally to the work. Collaboration is only acceptable between members of the same pair-team; inter-team collaboration is forbidden and violators may be sanctioned. Collaboration on examinations is not permitted; each exam must be completed by the individual student. Failure to abide by these rules will result in a score of zero being assigned to one or both members of the team (i.e., if I have a reasonable hunch that one student did all of the work on an assignment and the other student simply put his/her name on it, then the student who did all of the work will receive the assignment score and the other student will be given a score of zero).

Note: See the [Assessment](#) section concerning how academic integrity violations are handled for MPL exercises, lab projects, quizzes, and exams.

You are encouraged to acquaint yourself with the [ASU Academic Integrity Policy](#)

Classroom Behavior

The [ASU Student Services Manual](#) (SSM 201-10) permits the instructor to withdraw a student from a course for disruptive behavior with a mark of W (withdrawal) or E (failure). Note that "disruptive behavior" is defined by the instructor, not by the University or the student. Violation of conventional and acceptable classroom behavior will result in the offender being asked to exit the classroom and notification of the offense to the Fulton Schools of Engineering's Dean's Office. A warning may or may not be provided.

Note that in general, you may sit where you wish. However, the instructor has the right to ask you to sit in a specific seat or move to a different seat at any time during the semester. In the past, I have moved students whom I suspected were cheating during an exam, and I will do so in this course if I believe you are looking at another student's paper or sharing answers during an exam.

Attendance Policy

There is a strong and well-established correlation between class attendance, learning, and performance; therefore, regular class/lab attendance and participation is expected. I (or in the labs, the TA) intend to begin class each day on time, and we expect you to be present and ready at that time. However, you are adults, and you (or someone who may or may not love you) are paying for your education, and ultimately, it is *your* education. If you want to squander this opportunity, then no gimmick we devise to try to get you to come to class and participate will be successful, so lecture attendance will not count toward your final course grade. That said, 20% of your lab percentage is based on lab attendance.

Requirements for Success in this Course

The instructor assumes that you are mature and responsible adults, that you are enrolled in this course because you wish to learn the material, that you will read any assigned readings before class begins, that you will come to class prepared to discuss the reading and ask questions, that you will complete the assignments to the best of your ability on time, that you will actively participate in class discussions, and that you will ask questions about material you find confusing. The instructor believes that college students must be actively involved in their own learning process, that they cannot just sit and listen to lectures and expect to learn the material, that one of the purposes of college education and the Arizona State University mission is for the student to self-develop skills such as problem solving, independent learning, critical thinking, and effective written and spoken communication. To succeed in this course you must:

- Be prepared for every class, attend every class, and pay attention.
- Read the textbook and any assigned readings prior to class.
- Begin and complete the MPL exercises well before the due date.
- Prepare thoroughly for and complete every exam.
- Do any additional exercises you must to understand the material.
- Work with a partner if you wish or if it helps you.
- Ask questions in class. If you do not feel comfortable asking the question in class, talk with me outside of class.
- If you do not complete an assignment by the deadline, complete it anyway later.
- If you miss points on an assignment or exam determine why your answer was graded incorrect and learn the correct answers.
- Seek help from the instructor, TA, grader, or the tutoring center before you are too far behind on your understanding of the subject.
- Read your email every day; I often send important announcements via email.
- Check the course website and Blackboard every day for new announcements, material, and updates.

The purpose of the lab is to permit you to practice writing C++ programs with a TA and UGTA's present to assist you. Learning to write computer programs is *very difficult*. Of course, like many endeavors, some students find it much easier than others. However, even students who find it extremely difficult and challenging can do some things to enhance their learning and perhaps earn a higher grade, including:

- Complete any assigned prelab exercises before the lab session.
- Be prepared for and attend every lab session.
- Work diligently in the lab session and do not screw around.
- Ask the TA and UGTA's for help when you are stuck or confused.
- Work with a partner if you wish or if it helps you.
- Even if you do not complete a lab project, or you earn a low score on one, complete it later.

Having said all that, I want you to know that I care about all of my students and their education. I want all of you to succeed, to feel you have gained something from the course, to have some fun in the process, and I will do all I reasonably can to assist you in your efforts!

Statement on Accommodations

The **Disability Resource Center** (480-965-1234; Matthews Center; email: disability-q@asu.edu) is the central location for students requiring accommodation. Any student requiring accommodation must contact and register with the Center before any accommodation requests can be granted by the instructor. If you require accommodation, please contact the Center as soon as possible so the instructor can work with you to ensure your success.

Kevin R. Burger

School of Computing, Informatics, and Decision Systems Engineering

Ira A. Fulton Schools of Engineering

Arizona State University, Tempe, Arizona, USA, 85287-8809

Email: [burgerk\[@-sign\]asu\[.dot\]edu](mailto:burgerk[@-sign]asu[.dot]edu), Phone: +1 480.965.1493, Fax: +1 480.965.2751

Last Updated: Sat 16 Aug 2014 1410