

Contents

Deep Dive Java + Android Bootcamp Cohort 16	2
Course description	2
Calendar	2
General	2
Details	2
Student learning outcomes	3
Instructors and coaches	4
Contacting instructors, coaches, and staff	5
Office hours	5
Pre-work	5
Student coding deliverables	6
Guided programming exercises and projects	6
Individual Android project	6
Capstone project	7
Quizzes	8
Exams	9
Extra credit	10
Standards for completion	10
Required points	10
Gates	11
Integrity and intellectual property	12
Late work	13
Attendance and participation	13
Excused and unexcused absences	13
No-shows	14
Punctuality	14
Participation and practices	14
Online operation	15
Success plans	16
Professional development activities, tech lectures, and labs	17
Materials	17
Websites	17
Textbooks	17
Video recordings	18
Hardware and software requirements	18
Using continuous backup/synchronization cloud storage	19
Server and network credentials	20
Copyrights and licenses	20

Deep Dive Java + Android Bootcamp Cohort 16

Course description

The *Deep Dive Java + Android Bootcamp* is a 12-week long immersive course, covering a range of Java programming topics, as well as more general software development and engineering topics: core computer science concepts; object-oriented programming; fundamental software engineering practices; version control concepts and practices; Java syntax and structure; the Java standard library; relational database management systems; enterprise-level development; Android app development; enterprise application deployment on the Linux platform with the Apache HTTP server; elements of Agile methodologies; Material Design principles; build automation and dependency management; and data structures and algorithms will be explored extensively through structured exercises and projects.

Along with the above, professional development (including résumé writing, building an online presence, portfolio website development, writing and delivering effective presentations, and building and leveraging business networks) is also a key element of the bootcamp.

Calendar

General

- Pre-work: 25 August–17 September
- Orientation: 15 September, 1:30–3:30 PM
- Instruction: 18 September–8 December, Monday–Thursday 8:00 AM–5:00 PM, Friday 8:00 AM–3:00 PM
- Roundtable presentations of capstone projects: 8 December, 8:00 AM–12:00 noon
- Graduation & project demos: 8 December, 12:30 PM–3:00 PM
- Holidays: 23–24 November (Thanksgiving)

(All times MT.)

The high-level timeline in figure 1 shows the anticipated general areas of focus over the 12 weeks of the bootcamp. Solid horizontal lines indicate a strong focus on the specified element in the given week, while dotted lines indicate a content area that is addressed during the given week, but is not a primary focus. Vertical marks indicate milestones or other due dates.

Details

The Brightspace course for this cohort (<https://mycourses.cnm.edu/d2l/home/145333>) includes a detailed course calendar (<https://mycourses.cnm.edu/d2l/le/calendar/145333>). All quizzes, exams, project due dates, and other bootcamp events are shown in the course calendar and will be announced in class. **It is**

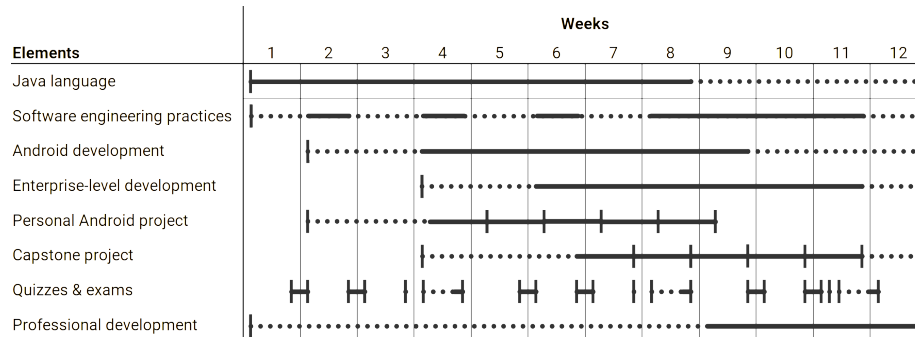


Figure 1: High-level timeline

each student’s responsibility to review the calendar regularly and to complete their exams and projects on time.

During the course, some event dates and due dates may need to be modified due to unforeseen circumstances. Any such changes will be announced in advance, and updated in the course calendar.

Student learning outcomes

Each cohort presents unique challenges and opportunities; thus, the course content will differ slightly from cohort to cohort. In general, students who complete the bootcamp will be reasonably prepared to:

1. Implement proven problem-solving techniques, incorporating core data structures and algorithms, as well as fundamental software engineering principles and practices.
2. Use Java 11 to build cross-platform console and GUI applications.
3. Write and run effective unit tests using the JUnit5 testing framework.
4. Use Git version control effectively to manage source code and other artifacts in individual and group projects.
5. Use industry-standard build and dependency-management frameworks to build, test, and package code as part of a CI pipeline.
6. Use Markdown (supplemented by HTML and CSS) with the Jekyll static site generator to create project documentation and personal portfolio websites.
7. Develop cross-platform database-backed services and applications using SQL, JDBC, and ORM technologies.
8. Connect to, transfer files to, and deploy Java enterprise artifacts to (Unix/Linux) server environments.

9. Perform basic Apache HTTP server configuration to set up virtual hosts and proxy connections to Java application servers.
10. Use the Android SDK, 3rd-party libraries, and DDC-developed application templates to develop Android apps incorporating XML layouts; basic UI controls; data-driven dynamic content controls; language and configuration-based localization; and access to external services.
11. As a member of a development team, build a multi-tier application incorporating an Android client app and Java enterprise-level components (relational databases, authentication, and Spring Boot-based REST web services).
12. Author multiple types of technical documentation, including Javadoc-based API documentation.
13. Participate in code reviews and scrum stand-ups.
14. Leverage their growing professional networks to identify and pursue employment and contracting opportunities.

Additionally, all concepts and techniques included in the Java SE 11 Developer exam (for the Oracle Certified Professional, Java SE 11 Developer certification) are covered in the bootcamp.¹

Instructors and coaches

The bootcamp has a staff of instructors with professional experience in software development, and in teaching programming concepts and techniques. Deep Dive also has coaches who work closely with students to maximize their chances of success in completing the bootcamp and in finding full-time employment, contract work, or internships after graduation.

- Lead Instructor: Nick Bennett
 - Slack: @nickbenn
 - E-mail: nbennett10@cnm.edu
- Student Success Manager: Sue Andres
 - Slack: @Sue Andres
 - E-mail: sandres1@cnm.edu
- Employer Relations Manager: Esteban Martinez
 - Slack: @Esteban Martinez
 - E-mail: emartinez394@cnm.edu

¹Oracle certification exams are administered under strict, timed conditions, with no access to outside materials permitted. Passing such an exam is not just a matter of having a solid grasp of the material, but also of committed practice and preparation on the part the individual. Thus, while the material will be covered in the bootcamp, and some in-class time will be dedicated to test-preparation resources and test-taking strategies, it is not possible for Deep Dive to guarantee that a student who has completed the bootcamp will pass the certification exam.

Contacting instructors, coaches, and staff

Outside of the classroom, the primary means of contact between bootcamp participants and instructors, coaches, and other bootcamp staff is Slack.² It's the participant's responsibility to contact the instructors for any issue related to the course—e.g. grading, completion of assignments, attendance, conflicts. It is also the participant's responsibility to check Slack on at least a daily basis (preferably more often) and to inform the instructors of any change in contact information or availability (e.g. inability to access the Brightspace course, Slack workspace, or GitHub organization for the bootcamp).

Office hours

Due to the intensive nature of the bootcamp—for instructors as well as students—regular office hours are minimal: An instructor will be available for meetings with students for 30 minutes after class, Tuesday through Thursday. Students are encouraged to make appointments; otherwise, walk-ins will be accommodated on a first-come, first-served basis.

Additional office hours may be scheduled, according to the instructors' availability and assessments of the need for such sessions. Typically, these are scheduled on one or more evenings or weekend days during the final sprints of the students' major projects. On the other hand, unforeseen circumstances will sometimes result in the cancellation or shortening of an office hours session; instructors will make a sincere effort to advise students of such changes at least 24 hours in advance, but this may not always be possible.

Pre-work

Prior to the start of full-time instruction, a number of pre-work tasks must be completed. These tasks fall into four groups, listed below; each group has a due date, by which all tasks in that group must be completed. The pre-work tasks are not graded, but they are treated as strict gates: Students must complete the tasks in each group by the deadline shown, or contact the lead instructor (by the same deadline) for assistance or to request an extension.

Task group	Due date
Getting connected	30 August
Environment preparation	7 September
Syllabus quiz	10 September
Basic tutorials	17 September

²The Slack workspace used for this bootcamp (as well as other Deep Dive bootcamps) is <https://deepdivecoding.slack.com>. Incoming students are invited to this workspace during pre-work.

Barring extenuating circumstances such as severe injury, illness, or a death in the student's immediate family, failure to complete a pre-work task group by its deadline, combined with failure to contact the lead instructor (by the same deadline) to request an extension, will result in the student not being permitted to continue in the cohort for which they have registered. The same outcome will result from failing to meet a deadline which has already been extended.

As implied above, instructors are available to support students who encounter difficulties or delays in completion of the pre-work; however, completion of the pre-work is ultimately the student's responsibility. In particular, success plans will not be created or used prior to the start of full-time instruction.

Details of the pre-work tasks are finalized three weeks before orientation. See the course content in Brightspace (<https://mycourses.cnm.edu/d2l/le/content/145333/Home>) for pre-work details and instructions.

Each student should plan to spend 40–60 hours to complete the pre-work tasks.

Student coding deliverables

Guided programming exercises and projects

Several applications are developed throughout the bootcamp by all participants, with the instructors' direct guidance. The conceptual, technical, and practical components of these development activities are designed to support the student learning outcomes.

Individual Android project

Each student will select (from a list provided by the instructors) and refine a topic for a standalone Android application development project, and then execute the development, starting from a provided application template.

Required, graded elements of this project are broken down into technical and documentation components as follows:

Technical components

- Multiple screens (activities or fragments), with intuitive, rational navigation.
- A persistent data store for runtime and configuration data. This usually involves a combination of Android Shared Preferences and the SQLite relational database.
- Dynamic, database-driven UI container components (e.g. `ListView`, `RecyclerView`, `ViewPager`, `GridView`).
- Custom graphic assets for launcher icons, button icons, etc.

Documentation components

- Design documentation, including wireframes and user stories.
- Technical documentation (e.g. entity-relationship diagrams, SQL DDL statements) for the persistent data store.
- Javadoc-based API documentation.
- Instructions for building, installing, and using the application.

While the bootcamp is not a writing course, an assessment of the student's ability to express themselves effectively in written form is part of the grading rubric for the project documentation.

Timeline The project timeline is divided up into 5 sprints of 1 week each, starting in week 4 and ending in week 9, with a milestone deliverable at the end of each sprint. The detailed list of deliverables and grading rubric for each sprint will be presented at the start of the sprint (or earlier).

Grading The grading of the project milestone deliverables (including the final deliverable) will be based on code and documentation committed to Git and pushed to GitHub by the students. There is not a minimum passing threshold score for each milestone; points earned simply add to the student's cumulative total for the bootcamp. However, submission of *at least some relevant work* for each milestone deliverable is a strict gate condition: Any student that fails to complete a project milestone will be placed on a *success plan*.

Capstone project

This is a multi-tier application created by a small team of students. Project topics will be selected by ranked-preference voting from a pool provided by instructors, and teams will be formed to execute the selected projects, starting from a DDC-provided template.

Required, graded elements of the capstone project include the following:

Technical components

- Persistent data stores for multi-user runtime and configuration data.
- Java implementation of server-side functionality.
- RESTful web services implemented in the server application and consumed by the client application.
- Java implementation of client functionality in an Android app.
- User authentication via OpenID Connect/OAuth 2.0 workflows.

Documentation components

- Design documentation, including wireframes and user stories.
- Technical documentation (e.g. entity-relationship diagrams, SQL DDL statements) for the persistent data stores.
- API documentation for all implemented web services.
- Javadoc-based documentation of client and server implementation code.
- Instructions for building and installing the client and server components, and for using the application.

While the bootcamp is not a writing course, an assessment of the team members' ability to express themselves effectively in written form is part of the grading rubric for the project documentation.

Timeline The capstone project timeline is divided up into 5 sprints of 1 week each, starting in week 6 and ending in week 11, with a milestone deliverable at the end of each. The list of deliverables and grading rubric for each sprint will be presented at the start of the sprint (or earlier).

Grading The grading of the project milestone deliverables (including the final deliverable) will be based on code and documentation committed to Git and pushed to GitHub by the team.³ There's not a minimum passing threshold score for each milestone: points earned simply add to each team member's cumulative total for the bootcamp. However, submission of *at least some relevant work* for each milestone deliverable is a strict gate condition: If a team fails to complete any one of the capstone project milestones, the team members will be placed on success plans.

Quizzes

In this bootcamp, quizzes are used to reinforce concepts and techniques learned through the in-class activities and projects and to help students prepare for subsequent exams. The quizzes are not used in computing the overall grade; *however, they do serve as gates*: Students are expected to take and pass quizzes at a minimum threshold level of 90% to continue in the bootcamp; multiple attempts are allowed.⁴

A total of 5–6 quizzes will be administered during the bootcamp, scheduled approximately every week in which no portion of an exam is administered. They

³Capstone projects are graded on a team basis: Unless a significant disparity is apparent, instructors will not attempt to assess individual contributions to the project.

⁴In addition to the quizzes administered during the course, a quiz on this syllabus is included in the pre-work. Each student must take this quiz as many times as necessary to earn a score of 100% before orientation.

will be made available to students at the end of class on Friday. For conceptual quizzes—administered in Brightspace, and consisting of multiple-choice, multiple-answer, true/false, matching, and ordering questions—successful completion is required by the start of class on the Monday after the quiz is made available. For practical quizzes—each requiring completion of a specific, constrained programming task—the deadline is the start of class on the Tuesday after the quiz is published.

Exams

There are 3 exams given during the bootcamp, administered approximately every 4 weeks, with each scheduled for a total of ~6 hours of class time. Each exam is organized in 2 parts, administered 5–7 calendar days apart:

- A set of multiple-choice, multiple-answer, true/false, and short-answer questions, similar in content and difficulty to the questions in the quiz (or quizzes) preceding the exam.
- A pool of practical programming tasks, from which students can select. These tasks generally take one of 3 forms:
 - Write a method to perform a specified processing task on inputs supplied via parameters. Additionally, write one or more test methods to demonstrate that the method gives the expected results with a set of provided test cases.
 - Write a console-mode or GUI application to take input data from the user, process it according to provided specifications, and display the results.
 - Given an existing application (possibly not yet building or running successfully), modify, correct, or complete the code following the stated technical and functional specifications.

Students are given information (without some key details) about a subset of these tasks in advance; they’re encouraged to research the problems and collaborate on possible approaches with their fellow students. However, work during the exams must be performed individually.

Grading of the practical exam problems is based on code committed to Git and pushed to GitHub by the students. Grading rubrics for these problems are distributed along with the problems.

These exams do not have a minimum passing threshold score; points earned simply add to the total points earned toward completion of the bootcamp. However, submission of *at least some relevant work* is a strict gate condition: a student that misses—or fails to submit work for—either part of any one of the 3 exams will be placed on a success plan.

Extra credit

At the lead instructor's discretion, there may be multiple extra credit opportunities offered during the bootcamp, potentially including:

- Exemplary execution of project elements.
- Completion of optional project elements
- Partial or full completion of optional practical exam problems, or optional elements of one or more practical exam problems.
- Partial or full completion of one or more standalone challenge tasks—typically declared with short notice, and with short windows for completion.

Since unanticipated circumstances may result in changes to project deadlines or grading rubrics, any given extra credit opportunity may be amended (within reason) after its initial announcement.

Standards for completion

To complete the bootcamp, a student must earn—through homework, project work, and exams—at least the minimum required point total and complete all gates.

In very restricted cases, exceptions to the standards stated here may be made:

- Extreme extenuating circumstances—which should be understood to include primarily severe injury, illness, or a death in a student's immediate family—may result in adjustments to due dates, gate conditions, etc.
- A student may work with CNM's Accessibility Services (<https://www.cnm.edu/depts/accessibility-services>) to request accommodations that could result in adjustments to gate conditions, due dates, time allowed for exams, etc. However, some requested accommodations may not be implementable for practical reasons, while others may be denied because they would compromise the student learning outcomes; in such cases, the student should work with Accessibility Services to identify other course options.

Required points

The overall score required to obtain a passing grade in the bootcamp is 750 points, which is 75% of the total points possible, broken down in the following table.

Component	Date due	Points available
Projects, exercises, homework		
Class projects, exercises, homework		200 ⁵
Individual Android project	14 November (final milestone)	200

Component	Date due	Points available
Capstone project	1 December (final milestone)	300
Assessments		
Exam 1	6 & 13 October	100
Exam 2	3 & 10 November	100
Exam 3	28 November & 4 December	100

Gates

All quizzes, exams, and project milestones are strict gates of this bootcamp, along with the final roundtable presentation of the capstone project. That is:

- All quizzes must be passed, each with a score of 90% or higher, within the time allowed, *regardless of the number of attempts*.⁶
- At least some relevant work must be submitted for both the conceptual and the practical components of all exams, within the time allowed for each, *regardless of the number of points earned*.
- At least some relevant work must be submitted by the individual student for all milestones of the personal Android project, and the student team for all milestones of the capstone project, within the time allowed for each, *regardless of the number of points earned*.
- The student must actively participate in the development and delivery of a final presentation of their team’s capstone project, *regardless of the state of the project implementation*.

After the start of full-time instruction, and before the start of the final week of the bootcamp, failure to complete any of the gates will trigger the creation of a success plan. However, in the final week, there is not sufficient time to create and complete a success plan; thus, failure to complete a gate in that week is especially serious: **Barring extreme extenuating circumstances or relevant accommodations, a student who does not submit relevant work on the final practical exam or who does not participate in the presentation of their team’s capstone project will not complete the bootcamp successfully.**

⁵Class projects, exercises, and homework are graded at the end of weeks 3, 6, 9, and 12. In addition to the points awarded for the work completed, participation in classroom exercises is included in the “Attendance and participation” section.

⁶In addition to the quizzes administered during the course, a quiz on this syllabus is included in the pre-work. Each student must take this quiz as many times as necessary to earn a score of 100% before orientation.

Integrity and intellectual property

A large part of the software developer’s work is research: Investigating possible approaches to a problem, exploring previous developers’ implementations, building and experimenting with trial solutions—even using an AI-assisted code generation tool—before finally selecting and focusing on a single implementation approach, and carrying it through to completion and successful testing. Recognizing this, we permit students to adapt code found or generated through such a process into their assignments, projects, and even exam work. However, strict conditions are placed on this use:

- **3rd-party code must not be copied (even if it is subsequently modified) without permission from the author.** In most cases, this permission doesn’t need to be requested explicitly but is granted implicitly under the specific conditions of an open-source license. (For example, code and other content contributed to StackOverflow is licensed for use under a Creative Commons Attribution-ShareAlike license.)
- **Any use must comply with all applicable license terms.** It is the student’s responsibility to identify all relevant licenses for any 3rd-party code they use and to complete the conditions for use under those licenses. This applies even to code that is modified from its original form.
- Even if the student incorporates code from the public domain, or if the applicable license does not require it, *this bootcamp requires attribution*: **All code taken or adapted from a 3rd-party source for use in this bootcamp must include attribution.**
- **License notices and other attributions must be included in the submitted work.** For exam or homework problems, license notices or other attributions must be included in comments accompanying the code; for project work, any such attribution must be included in the project documentation.

Failure to include the required license notice or other attribution when using code adapted from a third-party source will result in a grade of zero (0) for the student’s work on the homework assignment, exam problem, or project component in question. Repeated failures will result in a success plan being created for the student, with specific elements of the plan focusing on intellectual property.

Finally, keep in mind that indiscriminate copying and pasting of code—*without understanding the operation of the code, its side effects, and the result it returns*—rarely produces the desired outcome. In many cases, the resulting code won’t compile; in many others, the code will compile but will not behave as expected at runtime. Thus, students are expected to do the necessary work to understand the code they find or generate and adapt it to their purposes.

Late work

As a rule, full credit is not available for project work submitted late, and tests may not be completed after the due date.

Under extreme extenuating circumstances, or by prior agreement with the lead instructor, a student may be allowed to complete a test on a later date than that shown above, for full or partial credit (depending on the reason for the extension).

For projects, work will be graded based on the work submitted as of the date due. If a team (for the capstone project) or an individual student (for the individual project) submits a project deliverable late, there will be a penalty of 10% assessed for the first day (or fraction of a day) late, and up to 10% per day thereafter.

Attendance and participation

As stated in the course expectations and policies, attendance is crucial: persistent failure to maintain satisfactory attendance will result in a student's dismissal from the bootcamp. Of course, we recognize that emergencies and unexpected conflicts do occur; we will do our best to work with students to minimize the impact of unforeseen emergencies and conflicts on their success in the bootcamp. However, there are certain standards we must enforce. Most importantly, students are expected to attend and participate in at least 90% of the instructional (as well as team-building, professional development, etc.) hours of the bootcamp.

Excused and unexcused absences

An absence that is approved in advance by the instructors, or one due to an unforeseen medical emergency (for the student or a member of the student's family), with supporting documentation provided by the student after the fact, will be considered an excused absence. Any other absence will be considered unexcused.

The primary practical difference between an excused and an unexcused absence is that deadlines may be adjusted (as appropriate) for an excused absence, but there will be no such adjustment for an unexcused absence. (In either case, the student is expected to complete and submit any work due or assigned during the absence.)

In our experience, missing more than 10% of the in-class hours of a bootcamp makes it very difficult for a student to complete the bootcamp with the expected conceptual understanding and practical experience. Thus, missing more than 10% of the portion of the bootcamp that has elapsed to date, through any combination of excused or unexcused absences, will result in the student being placed on a success plan, and—if not brought back up to the 90% level by the ending date of the plan—will lead to the student's dismissal from the bootcamp.

Because an absence of a single day during the first two weeks will meet or exceed the 10% absence threshold, the lead instructor *may* choose to address this less formally, bringing the situation to the student's (and the Student Success Manager's) attention without creating a success plan. This option is most likely to be exercised if the absences in the first two weeks exceed 10% by only a small margin, and if there's a reasonable expectation that the deficiency can be corrected before the end of the third week.

No-shows

An unexcused absence of 4 or more hours, during which the student has not contacted the instructors regarding the cause and expected duration of the absence before 1 hour after the start of that absence, will be considered a *no-show*.

No-show absences are *very* disruptive—to the individual student's learning, to the student's capstone team's progress, and to the entire class's learning and morale. As a result, a student with no-show absences adding up to more than 8 hours in total will be placed on a success plan, even if the no-show absence occurs during the first two weeks of the bootcamp. **Additional no-show absences that occur while such a success plan is in effect will result in the student's dismissal from the bootcamp.**

Punctuality

Punctuality is expected of all bootcamp participants. Students are expected to be at their workstations, ready to begin, at 8:00 AM, and again at 1:00 PM. Similarly, other than scheduled breaks, students are expected to continue in the classroom until 5:00 PM (3:00 PM on Fridays), unless explicitly dismissed before that by the instructors.

If a student arrives 30 or more minutes late, departs the class 30 or more minutes early, or is otherwise absent from the class for a period of 30 minutes or longer, *without contacting the instructors*, it will be treated as an unexcused absence of $\frac{1}{2}$ day.

Even with instructor notification, late arrivals, early departures, or other unexcused out-of-class intervals totaling more than 1 hour in a single week will be treated as unexcused absences of that total duration.

Participation and practices

The standard for satisfactory attendance in this bootcamp is not simply a matter of physical presence, but also includes active participation in classroom lectures, exercises, code reviews, and quizzes; the final capstone project presentation; team-building and professional development activities (which have exactly the same attendance and punctuality requirements as the regular instruction sessions); and adherence to a basic set of practices, standards, and conventions.

In particular, the expected practices include the diligent, consistent use of the following—not just during the 12 weeks of formal instruction, but also during the pre-work:

- Slack, for intra-cohort and Deep Dive-wide discussions, capstone team coordination, and communication with staff and instructors;
- E-mail, for more formal, persistent communications with staff and instructors;
- Git version control for source code and other project artifacts;
- GitHub service for maintaining repositories of work done in the bootcamp.

Related to the last 2 items, note that the student’s work on classroom exercises, homework, projects, and practical exams is assessed *according to what has been placed under Git version control and pushed to GitHub*. Failure to make use of Git and GitHub as directed will result in the student receiving a score of zero (0) for the project, activity, or exam in question.

Persistent failure to engage with these tools and practices during pre-work will almost certainly result in failing to complete one or more required tasks, and subsequently to the student not continuing with the cohort. During the bootcamp itself, it will trigger the creation of a success plan, and—if not corrected—will lead to the student’s dismissal from the bootcamp for unsatisfactory participation.

Of course, it takes time, attention, and repetition to change practices. We don’t assume that any student will be instantly impeccable in their adoption of these tools and practices. So, we’ll emphasize them repeatedly in the classroom, and remind students of the expectations multiple times, before (and with the aim of avoiding) escalation.

Online operation

Please note that the standards for attendance, punctuality, and other practices outlined above apply even in online or hybrid modes of instruction—that is, when one or more students are attending remotely.

We recognize that online conferencing tools are never 100% reliable: internet connectivity and conferencing service interruptions are not uncommon; even multi-hour outages can occur. If a student is unable to connect or maintain a connection to a scheduled online activity, the student is expected to keep the instructors apprised of the situation (via Slack or e-mail) on a timely basis.

(We also strongly recommend that all students install Slack on their phones, to enable more fault-tolerant channels of communication with instructors and teammates.)

Success plans

The aims of the success plan process are to identify an at-risk student while there's still time to correct the situation, and to provide a clear path forward.

If a student is at risk of not completing the bootcamp—due to failure to complete a gate, a running point total below 75% of the points available to date, or unsatisfactory attendance or participation—the process of creating, coming to agreement on, and executing a success plan begins:

1. The student is advised that a success plan is being drafted by the lead instructor. Typically, the draft includes most or all of these elements:
 - A list of deficiencies triggering the creation of the plan.
 - Starting and ending dates of the success plan. (The duration of a success plan is usually 2–4 weeks.)
 - Actions (with due dates) to be completed by the student, lead instructor, and Student Success Manager.
 - A schedule of periodic meetings between the student and the lead instructor, and between the student and the Student Success Manager, to review completed and in-progress actions, and to identify and provide additional support as needed.
 - Unambiguous resolution criteria—that is, conditions that would result in satisfactory completion of the success plan, and conditions that would result in amendment or extension of the success plan.
2. The lead instructor's draft is reviewed by the Student Success Manager.
3. The student reviews the (possibly revised) draft, and may request modifications before agreement. Any such requests are discussed and resolved as quickly as possible. These requests aren't automatically granted, but *may be* (sometimes with additional modifications), as long as the requested changes wouldn't place an unacceptable additional workload on the instructors or the Student Success Manager, and as long as they wouldn't compromise the student learning outcomes.
4. After coming to agreement, the success plan is put into effect.
5. If the actions specified in the accepted plan are subsequently completed sufficiently to meet the resolution criteria, the success plan is closed satisfactorily; otherwise, depending on the actions completed, the success plan is extended or amended, or the student is dismissed from the bootcamp.

Barring extreme extenuating circumstances or a relevant accommodation, a success plan will not be created in the final week of the bootcamp, nor will a success plan have a completion date after the final day of the bootcamp.

Professional development activities, tech lectures, and labs

Cohort-specific professional development activities are conducted roughly every 1–2 weeks of the bootcamp. These generally include business networking events, a practice interview day, and the roundtable capstone project presentation on the last day of week 12. **Participation in these activities is required.**

On most Fridays, the usual Java-focused instruction ends at midday. The remainder of the Friday schedule varies from week to week, but can include presentations by Deep Dive alumni or local tech employers; “soft skills”-building sessions; community-building and networking activities with students from multiple bootcamps; lectures on development topics related—but not central—to the bootcamp curriculum; and so on. **Participation in these sessions is required.**

Materials

The materials used in this bootcamp are a combination of custom-developed instructional modules, tutorials, and coding problems; the textbooks listed below; video recordings of classroom sessions conducted in online or hybrid mode; and selected materials from outside sources.

Course materials other than the textbooks are provided as the activities or specifications detailed in or supported by those materials are assigned. Apart from video recordings and quiz or exam problems, these materials remain available to students for at least one year after graduation.

Websites

- The Brightspace classroom for this cohort (<https://mycourses.cnm.edu/d2l/home/145333>) is the main portal for the online content used in this bootcamp. Among other components, it has a detailed course calendar; because there may be slight changes while the bootcamp is in progress, students are expected to review the calendar frequently.
- The GitHub organization for this cohort (<https://github.com/ddc-java-16>) will host remote clones of students’ code repositories for in-class exercises; cohort, team, and individual projects; and practical exam problems.

Textbooks

Textbooks are distributed to students before or during orientation. E-book distribution is generally the default, but students may request physical books in some cases.

- Head First Java (3rd ed.)
- Android Studio Giraffe Essentials—Java Edition
- Effective Java (3rd ed.)

Video recordings

When any bootcamp participants are attending online (even when a student has been granted a waiver to attend an otherwise in-person bootcamp remotely), lectures and screen shares will be recorded to the Zoom cloud and subsequently made available to students via CNM’s OneDrive infrastructure. While the instructors will make every effort to have these recordings available online within 24 hours of the end of each day of instruction, this is not always possible. However, such delays in a recording’s availability online—or even the lack of a recording for one or more classroom sessions—**will not** be considered grounds for extension of assignment, exam, or project deadlines.

Recordings remain available for at least 30 days after the end of the bootcamp.

Hardware and software requirements

Participation in the bootcamp requires a laptop provisioned as follows:

- **Operating system (64-bit)**
 - Microsoft Windows 10 1809 (October 2018 update) or higher
 - macOS 11 (Big Sur) or higher
 - Ubuntu 20.04 or higher LTS release⁷
- **CPU**
 - 64-bit Intel, AMD, or Apple⁸
 - 2 cores minimum, 4+ cores **strongly** recommended
- **RAM**
 - 8GB minimum, 16GB+ **strongly** recommended, if primarily using a physical device as an Android test environment
 - 16GB minimum, 24GB+ **strongly** recommended, if the Android Emulator is used extensively⁹
- **Free/available HDD or SSD (preferred) storage space**
 - 30 GB minimum, 50+GB **strongly** recommended¹⁰

⁷Both the JDK and IntelliJ IDEA can run on many different versions of Linux. However, given the wide variety of Linux distributions, Deep Dive cannot be responsible for specification or support of bootcamp-ready configuration of any Linux distributions other than the Ubuntu LTS releases.

⁸While some Android Emulator configurations don’t require hardware-accelerated virtualization, these tend to run too slowly for practical development use. Thus, we *do not* support the use of Android emulation on CPUs that do not provide virtualization.

⁹Running IntelliJ IDEA alongside an emulator in less than 16GB of RAM usually results in very long Android app build times, frequent IntelliJ IDEA restarts, and severely limited functionality of the symbolic debugger in Android apps. As a result, we *do not* support using an emulator as the primary Android development target with less than 16GB of RAM.

¹⁰The disk space required for the tools, libraries, and projects of the bootcamp can vary widely from student to student. We **strongly** recommend maintaining at least 10%–15% free space on a fully-loaded drive—even after environment setup is complete.

The basic requirements for running the Java compiler, runtime, and IDEs used in the course are those listed above. In addition, you must have a suitable environment for running and testing Android apps. If you have an Android phone or tablet, you may use it for this purpose; alternatively, the Android Emulator component of the Android SDK may be used. The requirements for the Android Emulator can be found in “Emulator system requirements” (<https://developer.android.com/studio/run/emulator#requirements>).

The instructors will make every reasonable effort to assist you (as needed) in setting up one of these options; however, the wide variety of possible hardware and software configurations makes it impossible to guarantee that your preferred approach will be suitable for use in the bootcamp.

It is each participant’s responsibility to obtain and maintain their computing environment (including an Android test environment) in a working condition sufficient to complete the in-class activities, projects, and practical exams. In general, this requirement is easier to satisfy when the student starts out with a system configuration that not only meets the basic requirements stated above, but exceeds them.

Using continuous backup/synchronization cloud storage

In general, cloud storage platforms such as Microsoft OneDrive, Google Drive, Dropbox, or Apple iCloud, when running in continuous backup/synchronization mode, **are not well-suited** to directories containing a large number of frequently-modified files—e.g. Git repository metadata and object databases.

This issue is of particular concern for Windows users. By default, new Windows installations are configured to use Microsoft OneDrive; also by default, the **Desktop** and **Documents** directories are included in OneDrive cloud synchronization; thus, it is very easy to create a new directory in one of these 2 locations that has OneDrive synchronization enabled, *without the user even being aware of that fact*. The potential problems are compounded by the file- and directory-locking strategies employed by OneDrive: When OneDrive synchronization is not up-to-date, the user may not be able to modify the files in one or more directories until synchronization is complete. In the worst case, when OneDrive is unable to synchronize directory contents (e.g. if the storage space in a free OneDrive account is full), a user may be unable to change its settings to mitigate the issues, making the affected directories effectively read-only while the problem persists.

Given the above, and given the need to have as consistent a configuration as possible within each cohort, the environment preparation portion of the pre-work includes instructions for creating a directory, to be used for bootcamp projects and other content, in a location that is not usually included in the default configuration of these continuous backup programs. **We strongly recommend that this bootcamp content directory not be configured for continuous backup/synchronization with any of the above services**

(or any others). Further, any issues arising from conflicts with continuous backup/synchronization **will not** be considered grounds for deadline extensions on assignments, tests, or projects.

During the environment preparation and verification phase of the pre-work, instructors will make every reasonable effort to help students verify or modify their OneDrive configuration to avoid the issue described above, but it is the student's responsibility to ensure that their system configuration (including OneDrive and other cloud backup/synchronization services) is set up and maintained following the requirements of the bootcamp.

Server and network credentials

It's each student's responsibility to maintain access to all necessary server and network resources. Students are expected to record and maintain credentials for these resources in a reliable, secure credentials vault/password manager—**not** in an unencrypted file on the computer or in a physical notebook that's easily misplaced.

A loss of access to a required resource, due to lost/forgotten/hacked/locked account credentials, **is not** sufficient grounds for granting an extension on a homework, project, quiz, or exam deadline.

The credentials include (but are not necessarily limited to), the following, **all** of which are essential for successful participation in the bootcamp:

- CNM ID, e-mail address, and password;
- GitHub (<https://github.com>) username and password;
- JetBrains (<https://www.jetbrains.com>) username (or associated e-mail account) and password;
- SSH private key files and passphrases;
- Google (<https://www.google.com>) username and password;
- Hosting account usernames and passwords.

Several of the above encourage (or even require) two-factor authentication (*2FA*). Accessing 2FA-secured resources requires an additional step (such as entry of a one-time code sent by e-mail or text). It's the student's responsibility to maintain all required elements of the 2FA configurations (if any) for the platforms used in the bootcamp.

Copyrights and licenses

Except where explicitly stated otherwise, the copyright for all code written by the instructors for use in the course is held by CNM Ingenuity, and is licensed under the Apache License, Version 2.0 (<https://www.apache.org/licenses/LICENSE-2.0>).

The copyright for all non-code instructional material prepared by the instructors for use in the course is also held by CNM Ingenuity, with all rights reserved,

unless explicitly stated otherwise.

For their individual projects, students are encouraged (but not required) to license their work under an open-source license. For the capstone projects, licensing under an OSI-approved open-source license (see <https://opensource.org/licenses/alphabetical>) is required.