

CPSC 124 Introduction to Programming

About This Course

this course is about learning to write Java programs

```
public class HelloWorld {  
    public static void main ( String[] args ) {  
        System.out.println("hello world!");  
    }  
}
```

- a program is a list of instructions for the computer
- must learn both *syntax* and *semantics*
 - syntax addresses what is legal to write down
 - semantics addresses what it means i.e. what the computer does when it encounters that instruction

About This Course

but it is also about more than just learning to write legal Java instructions

a lot of computer programming is about problem solving

- **algorithm development** – figuring out how to perform a task in terms of a limited vocabulary of actions
- **coding** – writing down the instructions using the syntax of the chosen programming language
- **debugging** – finding and fixing mistakes in the instructions

also...

- program design
- establishing program correctness

About This Course

you're not going to be an expert programmer after just this course, but you will have a foundation

this course is about building blocks

- fluency in the basic syntax and semantics of Java
- solid grounding in the fundamental vocabulary of object-oriented programming
- strategies for selecting and combining programming constructs into programs that accomplish a desired goal
- good programming habits
- technical skills

- thinking like a programmer
- excitement about programming and computer science

About This Course

why Java?

- it is a well-known and well-established language – learning Java is a practical skill
- it is object-oriented – the dominant paradigm for many areas of programming including software development
- it has features that make it more friendly to beginners

About Computer Science

this course is the first course in the computer science major

computer science involves programming...

- CPSC 124 – Introduction to Programming
- CPSC 225 – Intermediate Programming
- CPSC 329 – Software Development

...but not just programming

- CPSC 220 – Introduction to Computer Architecture
- CPSC 229 – Foundations of Computation
- CPSC 327 – Data Structures and Algorithms
- electives which deal with applications of computing within various topic areas

About Computer Science

“Studying computer science means you'll be able to use computation in solving problems and that's a huge amplifier in our ability to advance any aspect of society.”

– John White, ACM CEO
(quoted in 12/10 CACM)

Course Materials

<http://math.hws.edu/bridgeman/courses/124/s24/>

CPSC 124: Introduction to Programming Spring 2024

Instructor [Stina Bridgeman](mailto:Stina.Bridgeman@hws.edu)
bridgeman@hws.edu
Lansing 302, x3614

Office Hours drop-in office hours: TBD or by appointment ([schedule](#))

Class Hours and Meeting Place lecture MWF 12:00-1:00pm - Stern 303
lab T 8:40-10:10am - Rosenberg 009

Course Links

- [Schedule](#)
(the course schedule, including links to handouts, assignments, reading material...pretty much everything you want on a daily basis is here)
- [Textbook](#)
(online copy of the textbook)
- [Course Information](#)
(course description, textbook information, required materials and software, assignments and evaluation, etc)
- [Course Policies](#)
(attendance, academic integrity and collaboration, late/makeup work, extensions, getting help, disability accommodations, etc)
- [Coding Standards](#)
(expected naming, formatting, commenting standards for your programs)

Documentation and Reference Material

- [Using Linux at HWS](#)

Schedule Page

check here for readings, assignments, handouts, examples from class, etc

CPSC 124 Introduction to Programming Spring 2024

CPSC 124 Schedule

Reading is to be done for the class period where it is listed; "Eck" refers to the textbook (Introduction to Programming in Java). Warmups are due by 10pm the night before the class for which they are listed.
Dates for things in light gray are for planning purposes and may be adjusted slightly.

Assignments

Week 1: 1/22-1/26
Topics: course introduction; the mental landscape; building blocks

Mon Reading:

- [Why is programming fun?](#)
- Eck sections 1.1, 1.3-1.5 (the mental landscape) [\[reading guide\]](#)
- (optional) Eck sections 1.2, 1.6-1.7 (asynchronous events, the modern user interface, the Internet and beyond)

Tue Reading:

- (recommended) Eck sections 2.1, 2.6.2 (the basic Java application, the commandline environment)

lab

Wed Reading:

- Eck introduction to chapter 2 and sections 2.1-2.2, 2.4.1, 2.4.6 (the basic Java application, variables and primitive types, text input and output) [\[reading guide\]](#)

warmup (on Canvas, under "Quizzes")

Homework

- complete the introductory survey (see the schedule page)
- consult the schedule page for the reading for Tuesday's lab
- consult the schedule page for reading and warmup for Wed (warmup due Tue 10pm)
 - we begin with the fundamental building blocks of a Java program
- review the policies and other information available on the course web page

<http://math.hws.edu/bridgeman/courses/124/s24/>

Course Materials

- textbook – *Introduction to Programming Using Java*
 - available online (link on course webpage)
 - can purchase a printed copy if you want
- software – Java programming language, Visual Studio Code, Linux operating system
 - available on the lab machines in Rosenberg 009 and Lansing 310
 - we will be piloting a Linux Virtual Desktop Interface, which allows you to access a Linux desktop from anywhere using a web browser
 - it is also possible to set up your own computer (optional)

Expectations

- attend all scheduled class and lab sessions
- spend approx. 8 hours per week outside of class on reading, assignments, and studying
 - you may need to spend more
 - if you routinely spend significantly less, you may not be sufficiently mastering the material

Assignments and Evaluation

key dates are on
the schedule page

- reading and warmups – first exposure to the material
 - reading is due the day it is listed
 - warmups due at 10pm on the day before – graded on effort rather than correctness
- labs and projects [50%] – opportunity for practice
 - lab assignments are started in lab, due one week later, will need to be completed outside lab
 - projects primarily completed outside lab
- syntax quizzes and skills checks [10%] – measure your progress mastering fundamental concepts
 - on Canvas
 - two attempts to successfully complete
 - additional attempts can be requested after office hours visit
- final exam and project [30%] – demonstration of mastery

3

Assignments and Evaluation

key dates are on
the schedule page

- engagement [10%] – your active participation in the learning process
 - warmups, attendance, participating in class, responding to “do you have any questions?” prompts, coming to office hours, ...
 - minimum expectation for a passing grade
 - complete 50% of the warmups
 - satisfy the attendance policy
 - for full credit, demonstrate greater engagement through some of the following –
 - completing more warmups
 - missing fewer classes
 - participating in class
 - regularly responding to “do you have questions?” prompts
 - coming to office hours

CPSC 124: Introduction to Programming • Spring 2024

14

Etiquette

- arriving late, leaving early, and coming and going during class is distracting
 - please do your best to arrive on time, and to take care of any necessary business before or after class so you can stay for the whole period
 - habitual late arrivals or early departures may be marked as an absence
 - let me know if you know in advance that you need to miss part or all of a class

CPSC 124: Introduction to Programming • Spring 2024

15

Policies

- attendance
 - this is an in-person course, and being in class and lab is an important part of the course
 - students who regularly miss class/lab often do not do as well
 - if you must miss class or lab, it is your responsibility to be aware of and make up missed content
 - check the schedule page for material from class and new assignments
 - come to office hours if you have questions
 - in addition –
 - more than 6 missed classes or 2 missed labs will lower your engagement grade
 - 4, 5, or 6 missed classes or 2 missed labs will lower your engagement grade unless you are proactive about managing your absences and demonstrate engagement in other ways
 - plan ahead when an absence is known about in advance
 - take steps to promptly make up missed content
 - communicate when you'll be absent and what you are doing to catch up
 - do warmups, participate in class, come to office hours, ...

Policies

- there is a steady workload, and sometimes multiple things in progress at one time – late handins can quickly snowball into falling behind
- the material is cumulative – every topic is built on and used for the rest of the course, so it is not possible to catch up by skipping ahead to the next topic
- late policy
 - partial credit for warmups handed in after 10pm but by class time, no credit after that
 - additional attempts can be made for syntax quizzes and skills checks after the due date as long as one attempt has been made before the due date
 - in general, labs and projects will not be accepted late
 - a late handin for an assignment completed on time (as established by timestamps) will be accepted, with a small point penalty if the late handin was due to a handin error (including forgetting to submit)
 - no work will be accepted after the end of the final exam timeslot
- extensions
 - no extensions for warmups
 - a two-day extension can be requested for labs and projects (one per assignment), *but it must be requested before the due date*
 - if you need more than an occasional extension or there are circumstances outside your control which are significantly impacting your ability to focus on academics for several days or more, *reach out promptly to discuss a plan for getting back on track*
 - no work will be accepted after the end of the final exam timeslot

7

Policies

- exams
 - if you have an unavoidable conflict with the date, see me in advance to discuss rescheduling
 - if a last-minute illness, personal or family emergency, or other crisis outside of your control means that you can't attend class on an exam day, contact me as soon as possible
 - otherwise exams cannot be rescheduled or made up if missed
 - the final exam can only be rescheduled in consultation with your dean

CPSC 124: Introduction to Programming • Spring 2024

18

Policies

- academic integrity and collaboration
 - syntax quizzes, skills checks, the final exam and project demonstrate mastery
 - solely your work, and only those resources explicitly authorized in the instructions
 - labs and projects for practice and learning – for *your* mastery
 - should first attempt on your own, using the course materials (textbook, posted slides and examples, linked references)
 - the primary resources for help or to discuss ideas should be office hours and Teaching Fellows
 - can discuss general topics (e.g. "how do for loops work?") with other students, but you may *not* discuss details of your programs with other students, work collaboratively with other students to write code for an assignment, copy someone else's solution, or use someone else's solution "as a guide" for your own
 - warmups are about self-assessment
 - the point is to identify your level of understanding – it should be solely your work
 - discussion with others after you've submitted your answers is encouraged
 - graded on effort rather than correctness

9

Policies

- academic integrity and collaboration
 - copying part or all of someone else's program is expressly prohibited, and it is not acceptable to be in possession of someone else's program before you have handed in your own
 - unless otherwise prohibited, other materials (such as reference books or websites) may be used as technical references to learn about a particular topic, however **looking for and/or copying a solution is not acceptable and you may not incorporate code you find in other materials or on the Internet into your program**, even if you make modifications to it
 - learning from examples is valuable – with an example, you have to understand it in order to be able to apply what it is illustrating to constructing your own solution
 - with solutions, you take someone else's work largely as-is and do not need to understand much about it
 - using someone else's program "as a guide" for your solution is still plagiarism even if there's a right answer and anyone solving the problem would arrive at a similar result

CPSC 124: Introduction to Programming • Spring 2024

20

Policies

- academic integrity and collaboration
 - AI systems (e.g. ChatGPT, Codex), homework help or study aid sites (e.g. Chegg, Course Hero), and sites where you post a homework problem or question and solicit answers from others **may not be used in the completion of graded work**
 - use of generative AI for study purposes is discouraged because it is an unreliable source

Being Successful

- stay on top of the material – things will pile up quickly if you fall behind
 - regularly utilize office hours and Teaching Fellows
 - start assignments early
- be proactive if you must miss class, especially if you miss several classes in a row, or if there is an ongoing issue which adversely affects your work
- ask questions / come to office hours and Teaching Fellows – don't wait if something is confusing or doesn't make sense
 - office hours are drop-in – no appointment needed
 - email and/or make an appointment if you can't come to office hours
- don't only ask "what do I do?" but also "how do I know that's what to do?"
 - the particular program is not the point of assignments – it's about the *process* by which a program is created to achieve a particular goal

Being Successful

- helpful practices
 - let me know if there are things about how the class is run that would be helpful for your learning
- disability accommodations
 - see the syllabus statement from Disability Services (on the Policies page)

The Mental Landscape

- a *program* is a list of instructions meant to be followed mechanically – that is, without thinking or understanding – by a computer
 - a *programming language* defines the *syntax* and *semantics* for a set of instructions
 - successful execution of a program requires that it have perfect syntax and unambiguous semantics, and that all details of the instructions are complete and correct

The Mental Landscape

- program instructions manipulate data
 - *variables, types, and assignment statements* deal with the storage of values
 - *operators and expressions* allow the computation of new values
 - *control structures* define the order in which instructions are executed
 - loops allow repetition, branches allow choice
 - *subroutines and classes* support modular programming, essential for creating a program of any size

The Mental Landscape

```
1263
+ 2458
-----
```