

Resume Best Practices

If you are reading this, the following is probably true:

- Bad news: Your resume has lots of room for improvement.
- Good news: If you think your resume is good now, it's going to blow your mind later.
- Great news: 90% of the issues / areas of development are common, and this guide / checklist will help you and your peers work through them.

Objective: To present yourself as an outstanding candidate

Again and again, we hear employers tell us they're looking for someone who:

- Is very strong at XXX and also has some experience with YYY
- Has some prior development experience
- Has a technical area of study
- Might not have prior experience, but is really talented and generally a badass

Working with many grads in the past and knowing the quality of our students has only gone up, we can tell you that you are most likely an outstanding candidate. The ultimate objective for our resume review & revision activities is to make sure you look that way as well on paper – i.e. to draw out as much of the above as possible and present it in a powerful and readable way.

Things your resume should scream “Yes!” to:

Fact: Readers will probably skim your resume for no more than 30 seconds to decide – “should I talk to this candidate?” There are three key contributing factors:

1. Is this person relevant?

- Is this person strong with the technical skills I need?
Bad response: Can't easily figure that out. Looks like a laundry list.
How to win: Make it easy to find and understand what you're strong at.

2. Is this person good?

- What is his/her depth of experience in software engineering?
Bad response: Looks very thin. I don't understand this (I'm not technical)
How to win: Draw out any work broadly related to software development, including non-paid work, side projects, etc. Of course talk about any work involving programming but also where you've built “software type tools (e.g. VBA, excel tools, etc.)
- What other kind of work has he done in the past? Is it badass?
Bad response: Weak experience. Not exciting. Whatever.
How to win: Accomplishments trump activities. Qualify & quantify everything. Do not allow a single bullet point to not read as badass.

- What kind of education has he had?

Bad response: What is Hack Reactor?

How to win: Describe HR as an “Advanced software engineering immersive or indicate the area of study as “software engineering”.

3. Is this person interesting?

Bad response: Person seems one-dimensional; not particularly memorable or eye-catching.

How to win: If you think your resume could use a boost, include a personal section (2-3 lines max) with interests & fascinating facts

Best Practices Checklist

Your resume should generally adhere (but does not have to be limited to) the following guidelines. Create a version of your resume that satisfies each of the points below, and then ask a peer to review your resume against the same checklist.

When conducting a peer review, one person should **always** be taking notes.

General – For Starters

- **Standard format:** Use a standard resume format; nothing fancy – it will maximize readability & ensure compatibility with applicant / resume tracking software (*check out old alum resumes*)
- **Text selectable PDF:** When PDF'ed, text should be selectable
- **Length:** Resume length should not exceed one page
- **Looks good:** Should look professional from a design standpoint – consistent font sizes, consistent formatting, reasonable page margins, etc.
- **Sections:** Content should be grouped with the follow headings / sections and ordered as follows (name can be named slightly different. A brief profile to start is OK, as long as it is unique as opposed to generic and does not exceed 2 lines in length):
 - If you have less engineering experience:
 1. Technical Skills
 2. Recent Projects
 3. Experience
 4. Education
 5. Personal
 - If you have more engineering experience:
 1. Technical Skills
 2. Recent Projects
 3. Relevant Experience
 4. Other Experience
 5. Education
 6. Personal
- **Avoid white space:** Avoid too much unused white space (*Goal: give the impression that you have tons to talk about but were unfortunately limited by space*)

Header

- **Github:** Github link should be visible on header
- **Raw hyperlinks:** Write out links – readers may have the resume printed
- **Address:** Your stated address should be where you are looking for work (*note: makes it clear they will not have to relocate you*)
- **Keep it tight:** Limit the amount of space this section uses, not more than 3-4 lines max.

Technical Skills

- **Indicate competency:** Skills should be grouped by level of competency –e.g. “strong vs.

knowledgeable vs. basic”, or “proficient vs knowledgeable” (*note: employers hate seeing a “laundry list” of skills. Also, avoid using the word “expert” as that often misrepresents things*)

- **Keep it tight:** Each skill should be a few words max, comma separated, grouped by category if desired. Avoid paragraphs (*note: showing what you know, i.e. through the experience section, is better than talking about what you know, i.e. this section*)

Recent Projects

- **Talking about projects:** For experience blocks about a project, the following structure / flow is recommended:

[Project Name]

Year (note: no months, right aligned)

- What specifically did you build? What was your role on the team? Did you work with another company / organization, in what capacity and how?
- What was the tech stack? Why did you choose those technologies? Why did you choose to architect your project in the way you did?
- What was a notable technical achievement or challenge you had to overcome that you think an employer would appreciate? (aim for at least 1 bullet point)
- What was the outcome of the project? What indicates you rocked it? (aim for at least 1 bullet point). Examples:
 - Deployed the app
 - Launched beta
 - Peaked at X unique users / day
 - Gained user base of X
 - App was successfully able to do X
 - Accepted by client
 - Client was interested in continuing work with you
 - Demoed to X people
 - Featured on X publication. Front page of Reddit or Hacker News
 - Planning to do X in the future
- **Representing projects:** To err on the side of clarity, we'd like all students to place all non-professional projects (such as HR projects) its own distinct section called "Relevant Projects", "Recent Projects", "Software Engineering Projects" or similar.
- **Representing project partners:** Please avoid the appearance of having an employment relationship with host companies / external partners if none exists
 - Instead of stating the host company as the title of your experience, please state your project name as the title - you can indicate the host company within your bullet points. For example, instead of suggesting that you worked for Famo.us, say you worked for [Project Name] and in your bullets you can say that you developed the project in collaboration with Famo.us as part of their Labs program.
 - If you would like to represent your experience differently, please seek explicit and written permission from the company you are working with.

Not Recommended

RELEVANT EXPERIENCE	
Host Company (e.g. famo.us)	2014
- Built....	
Greenfield Project Name	2014
- [bullets]	
XYZ Company, Position X	2013
- [bullets]	

Recommended

RECENT PROJECTS	
Thesis Project Name	2014
- Built... in collaboration with famo.us	
Greenfield Project Name	2014
- [bullets]	
PROFESSIONAL EXPERIENCE	
XYZ Company, Position X	2013
- [bullets]	

Experience – In general

- ❑ **Chronological order:** Experience blocks should run in chronological order, with the most recently completed experience appearing first
- ❑ **Recent experience only:** Experience from >10 years ago is too old to appear on your resume. It should either be removed or greatly downplayed
- ❑ **Don't toss out experience:** Thin on relevant experience? Students often make the mistake of tossing out experience they deem “irrelevant”. Here's the deal – what's relevant is that you are badass, so include all of your experience and portray it as awesome. Did something related to software development? It should, without question, be included:
 - ❑ Work experience, paid or unpaid (i.e. including volunteer experience)
 - ❑ Academic and research projects, especially if they are related to software development
 - ❑ Side / personal projects, especially if they are related to software development

Experience – For each block

- ❑ **Years:** Always indicate dates for each experience block, but without months. For example:
 - ❑ If start date is 2014 and you are still working on it, write “2014 - present”
 - ❑ If both start date & end date is 2014, write “2014”
 - ❑ If start date is 2013 and end date is 2014, write “2013-2014”. If your experience was very short (e.g. several months), avoid misrepresentation by just writing the end year
- ❑ **Break down large blocks of experience:** If one experience block covers many years of experience (e.g. 4+ years), consider splitting it out into two blocks, either by role (typically your role will have evolved over those years) or by project.
- ❑ **Bullets & verbs, not paragraphs:** Experience should be conveyed through bullet points starting with a powerful & meaningful verb, not paragraphs (*note: when skimming down a resume, you latch on to the first 1-2 words per line. This maximizes readability and content being conveyed*)
- ❑ **3-4 bullets per block:** 1-2 bullets is too little unless you are intentionally diminishing the opportunity, 3 to 4 bullets is OK, 5 could be stretching it, 6+ is generally too many bullets
 - ❑ Technically relevant and badass experience should be more lengthy vs. others
- ❑ **Badass bullets first:** Where possible, you should have the most relevant and impressive bullet points be bullet #1 and go downwards.
- ❑ **1-2 lines per bullet:** 1-2 lines per bullet is optimal. A bullet with 3 lines is acceptable if you are talking about something awesome and very meaty, but typically that would be too lengthy
- ❑ **Avoid hanging lines:** Avoid hanging lines by either adding words (to complete the line) or

removing words (to reduce one line)

The level of awesomeness conveyed by the presentation of your experience is almost always the #1 area for potential improvement.

Have a peer talk through each project / experience with you.

What are you most proud of? Why are you awesome? Why was your work a big deal?

- **Talking about each experience:** The following flow is recommended:

[Company / Org Name], [Role Name]

Year (note: no months, right aligned)

- Can't remember what you did? Run through your memory slowly and chronologically. Write out activities / projects / whatever comes to mind. That should be a good memory aid
- Use bullet points, starting with a verb, talk about accomplishments (over tasks / activities), outcomes and impact. Every single bullet should make the reader go "woah" and give the impression you were badass, vs. just doing your job. Here are some tips:
 - **Qualify:** What was the significance of your work, your role, your team or your company? Why it is a big deal? (e.g. Series A startup advised by X, one of the top 20 hospitals in the U.S., clients included Coca Cola, etc)
 - **Quantify the scope:** How many people, how much money, what volume of activity were you interacting with?
 - **Quantify the impact:** What were the outcomes of what you did? E.g. reduced work effort by X%, used in over 100 projects a year, increased accuracy by X%, etc. Can you demonstrate you were a top achiever via recognition (e.g. award, promotion, quote from boss) or statistics (e.g. top X% performer)?
- **Maximize technical relevance:** Search your memory for anything directly or peripherally related to software development and consider including it. Often, students have developed "tools" in a broad sense in the past – this is great to highlight. Talk about what the tool does and the impact

Education

- **Keep it tight:** You should not have more than 1-2 lines on each institution you attended. Ideally, the whole section should not exceed 6 lines – there's many better uses for space
 - Toss out names of awards & scholarships and just write "academic award recipient"
 - If you took courses related to software engineering, note "relevant courses: XXXX"
 - To save space, points under each institution can be period separated and kept on the same line vs. separate bullet points.
- **Describing Hack Reactor:** Describe HR as an "advanced software engineering immersive" or indicate the area of study is "software engineering" (otherwise people won't know what it is!)

Personal (optional – recommended if your resume can use a boost)

- **Humanize your resume:** Consider including a personal section (2-3 lines max) with interests &

fascinating facts about you. Keep each fact brief and period separated.

Example: Avid cook; consistently cooks for more than 150 friends a year. People tell me I make the best cornbread. I also enjoy standup paddleboard, rock-climbing, kayaking, and badminton.

This is awesome for giving your resume more color and to help the reader / interviewer relate to you. Definitely include both professional & casual but keep it classy :) Some ideas:

- ☐ Do you have another side to you (e.g. musician, chef, car enthusiast, poker player, etc.)?
- ☐ What are the top 2-3 things you like to do on vacation or in your spare time?
- ☐ Do you know multiple languages?
- ☐ Do you travel a lot?
- ☐ Did you have any interesting life experiences or life events?
- ☐ What are some fascinating facts about you? If you were creating a dating profile, and you wanted people to know you were interesting, what would you mention?