Name: Khushi Nitinkumar Patel

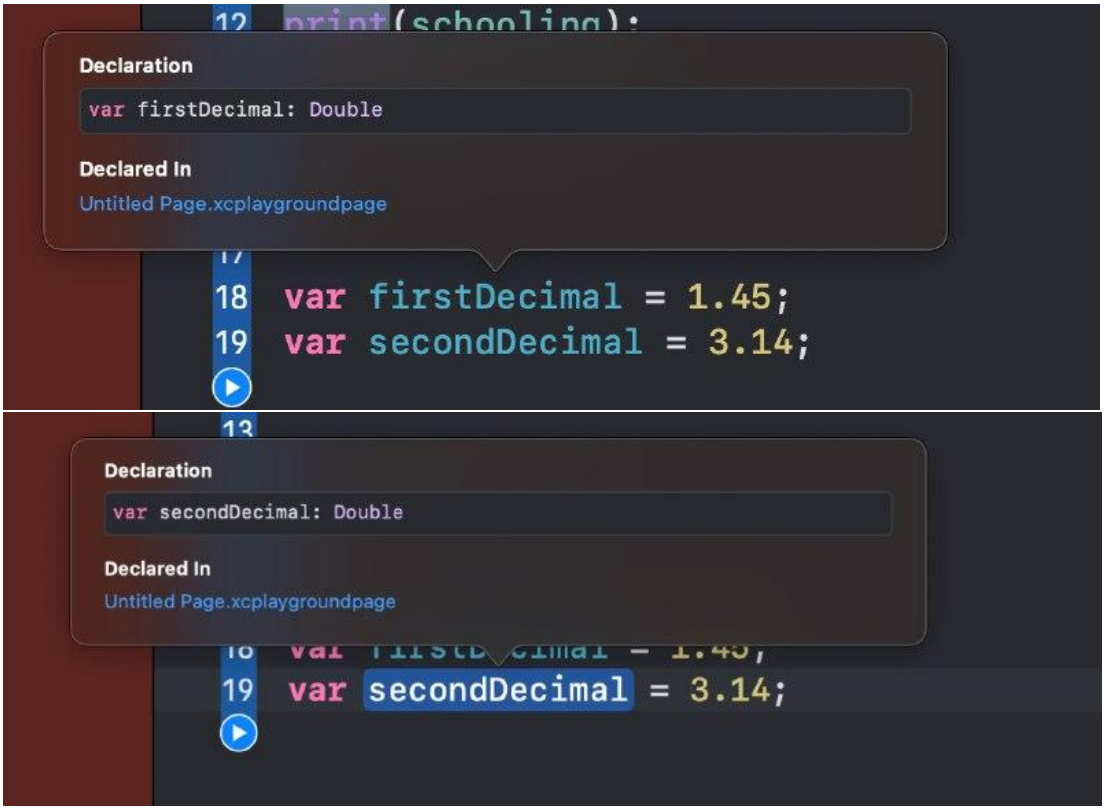
PRN: 2020BTECS00037

Batch: T2

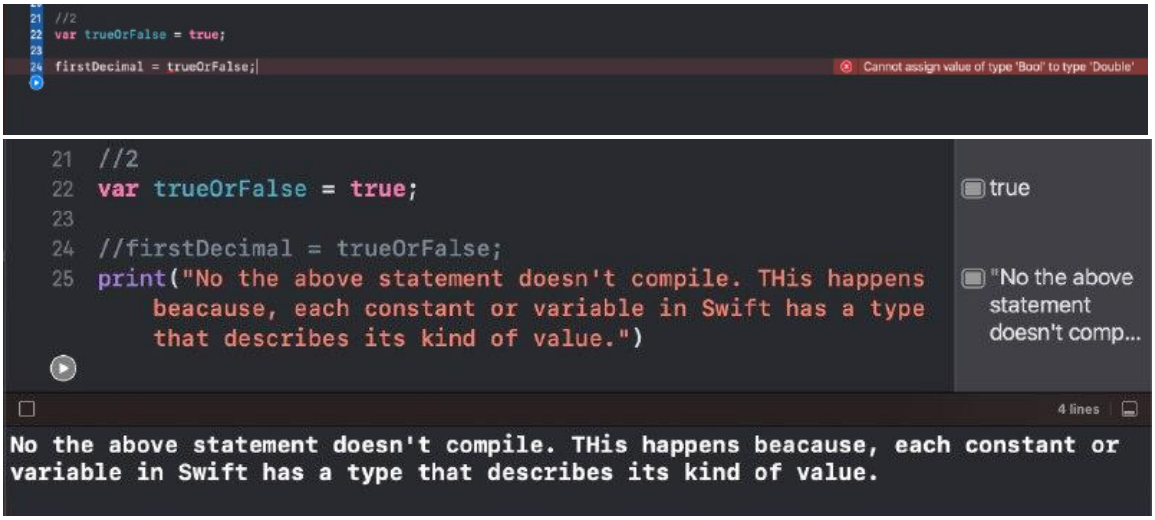
Assignment No 2

Types and Type Safety

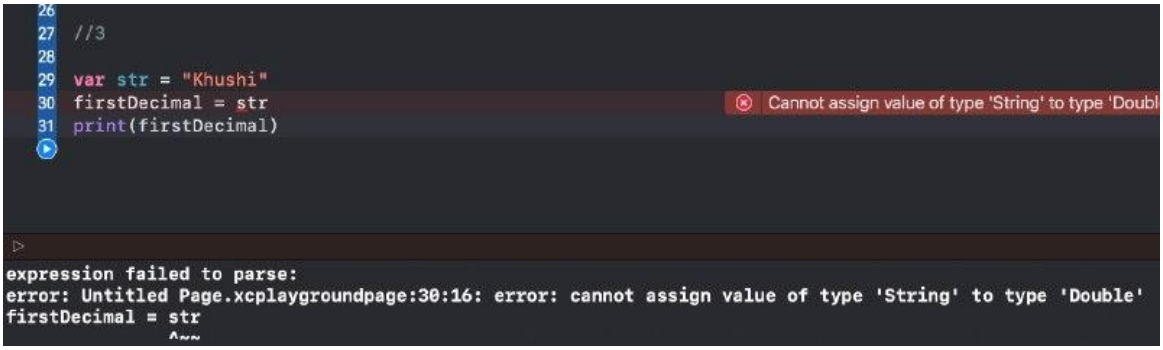
1. Declare two variables, one called `firstDecimal` and one called `secondDecimal`. Both should have decimal values. Look at both of their types by holding Option and clicking the variable name.

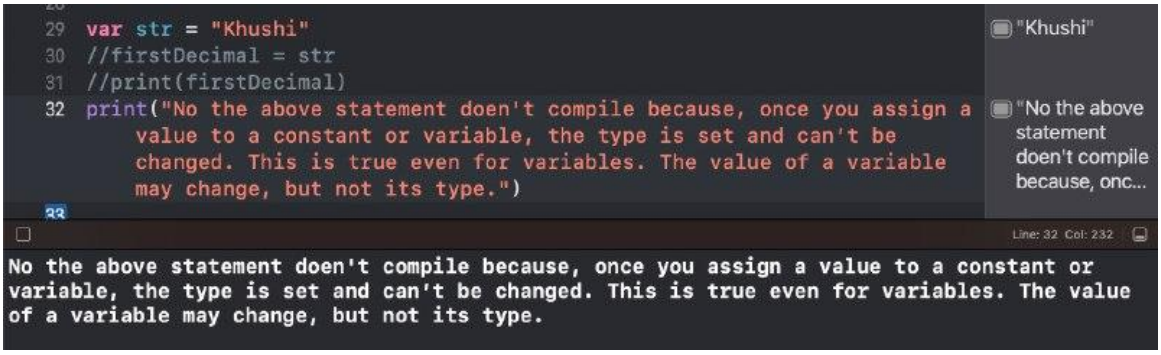


2. Declare a variable called `trueOrFalse` and give it a boolean value. Try to assign it to `firstDecimal` like so: `firstDecimal = trueOrFalse`. Does it compile? Print a statement to the console explaining why not, and remove the line of code that will not compile.

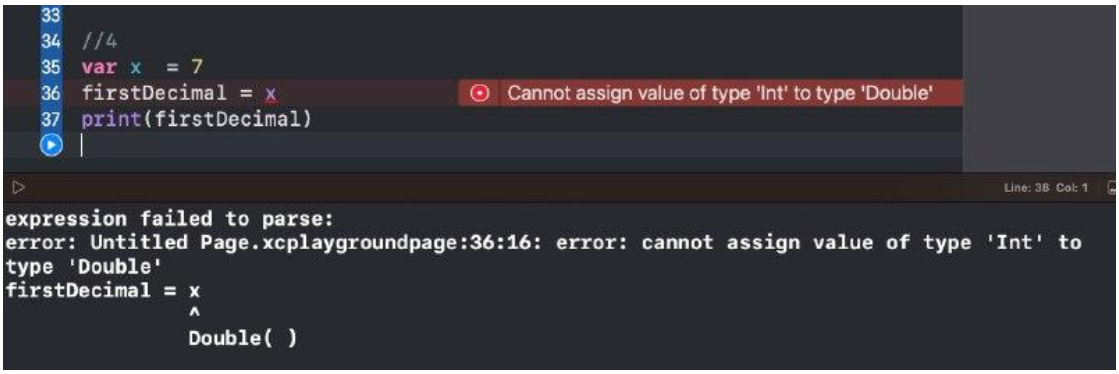


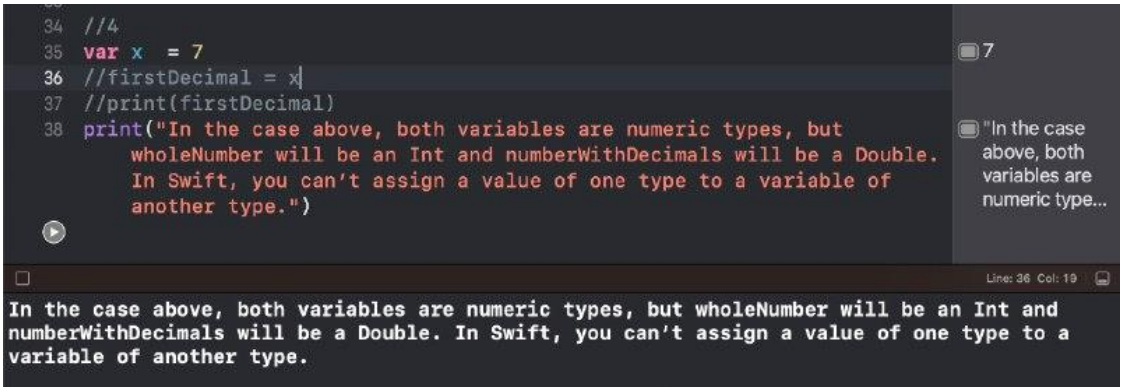
3. Declare a variable and give it a string value. Then try to assign it to `firstDecimal`. Does it compile? Print a statement to the console explaining why not, and remove the line of code that will not compile.



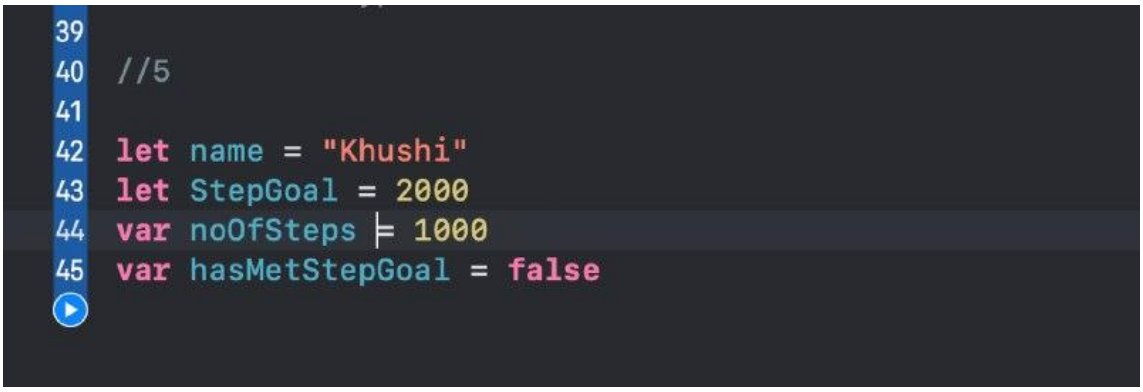


4. Finally, declare a variable with a whole number value. Then try to assign it to `firstDecimal`. Why won't this compile even though both variables are numbers? Print a statement to the console explaining why not, and remove the line of code that will not compile.





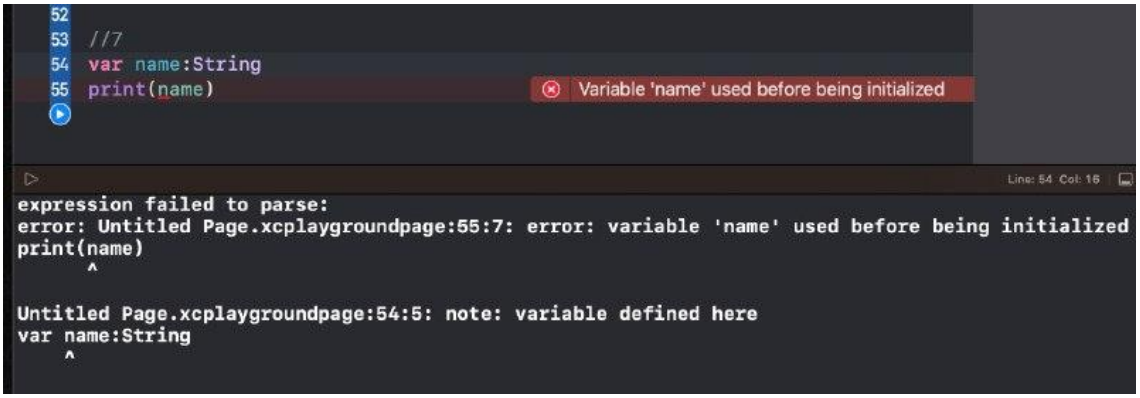
5. You have declared a number of constants and variables to keep track of fitness information. Declare one more variable with a boolean value called `hasMetStepGoal`.



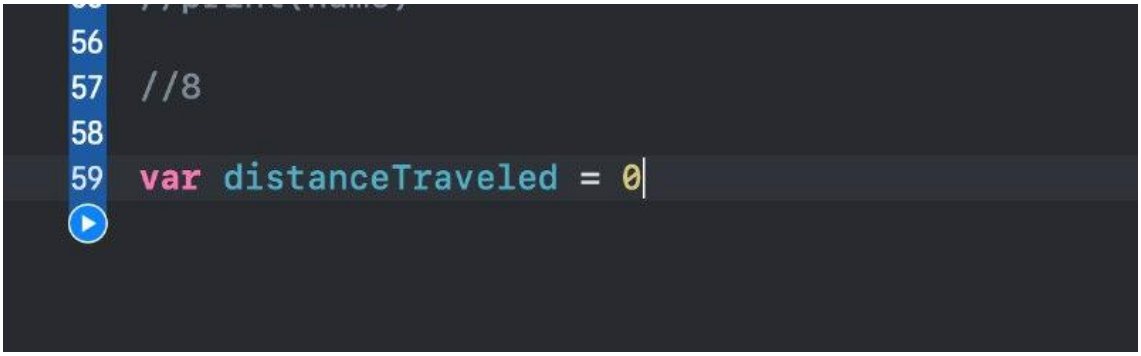
6. When you declared a constant for goal number of steps and a variable for current step count, you likely assigned each a value in the thousands. This can be difficult to read. Redeclare this constant and variable and, when assigning each a value in the thousands, format the number so that it is more readable.

Type Inference and Required Values

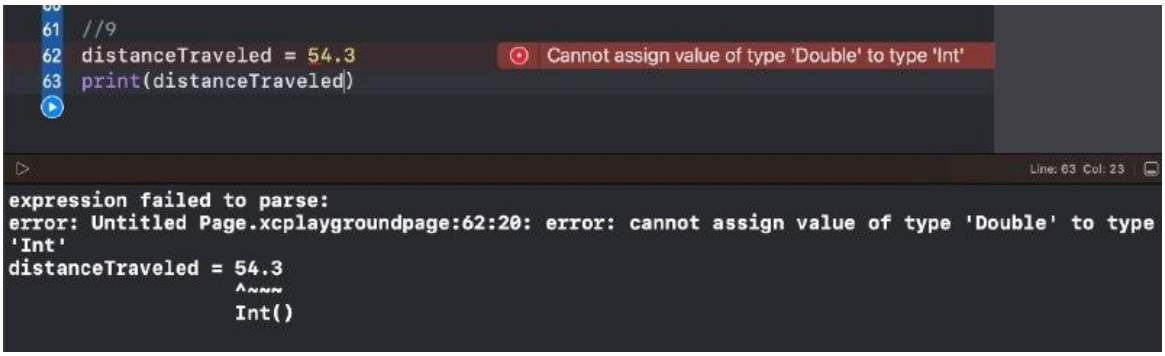
7. Declare a variable called `name` of type `String`, but do not give it a value. Print `name` to the console. Does the code compile? Remove any code that will not compile.

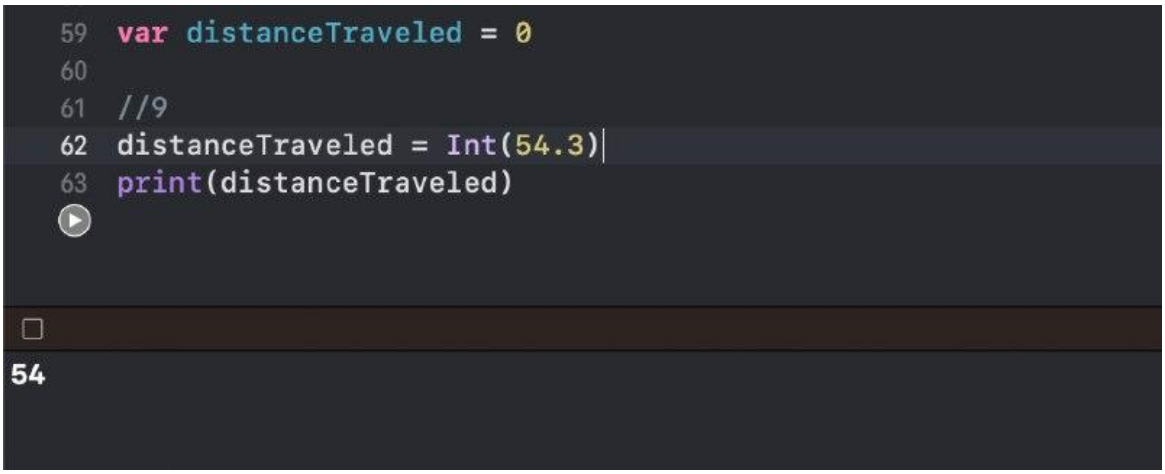


8. Declare a variable called `distanceTraveled` and set it to 0. Do not give it an explicit type.

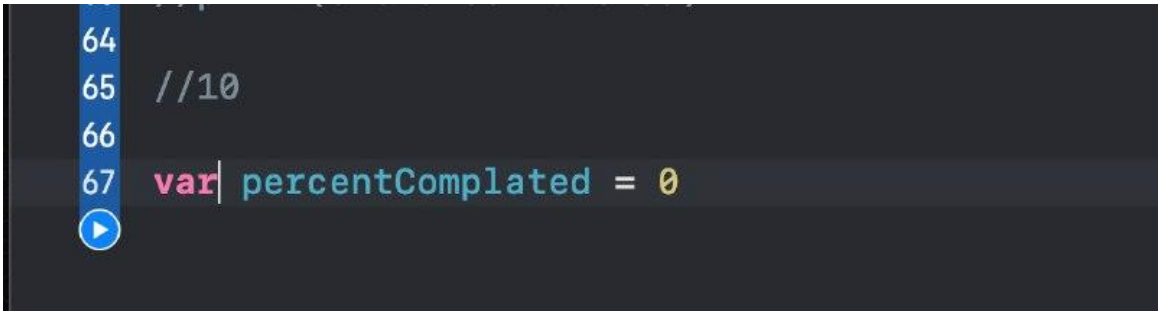


9. Now assign a value of 54.3 to `distanceTraveled`. Does the code compile? Go back and set an explicit type on `distanceTraveled` so the code will compile.

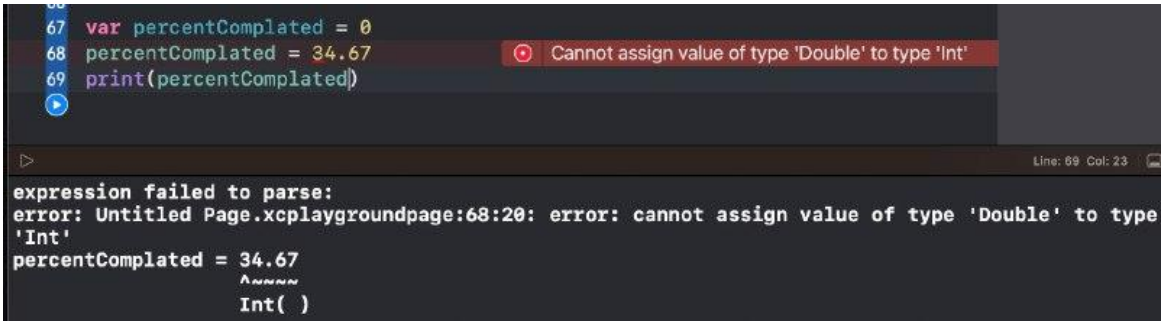


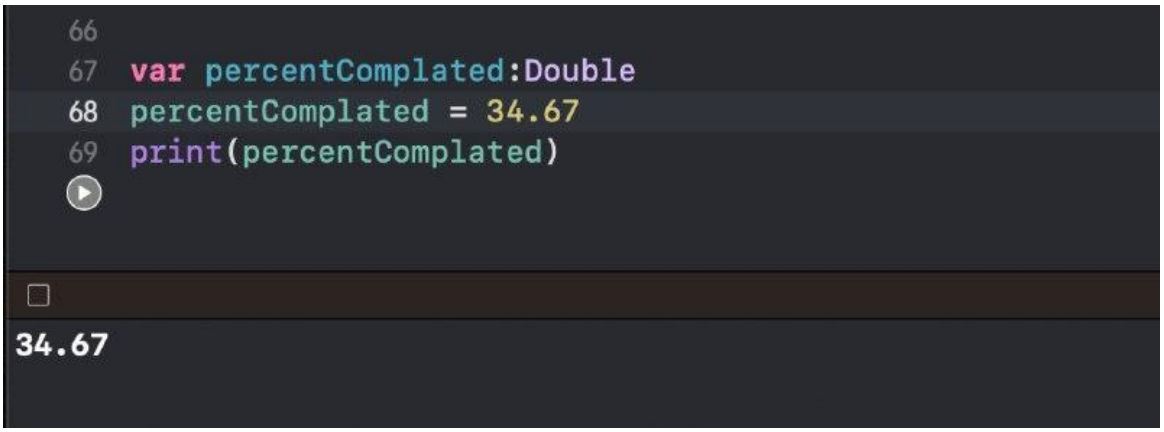


10. You decide that your fitness tracking app should show the user what percentage of his/her goal has been achieved so far today. Declare a variable called `percentCompleted` and set it to 0. Do not explicity assign it a type.



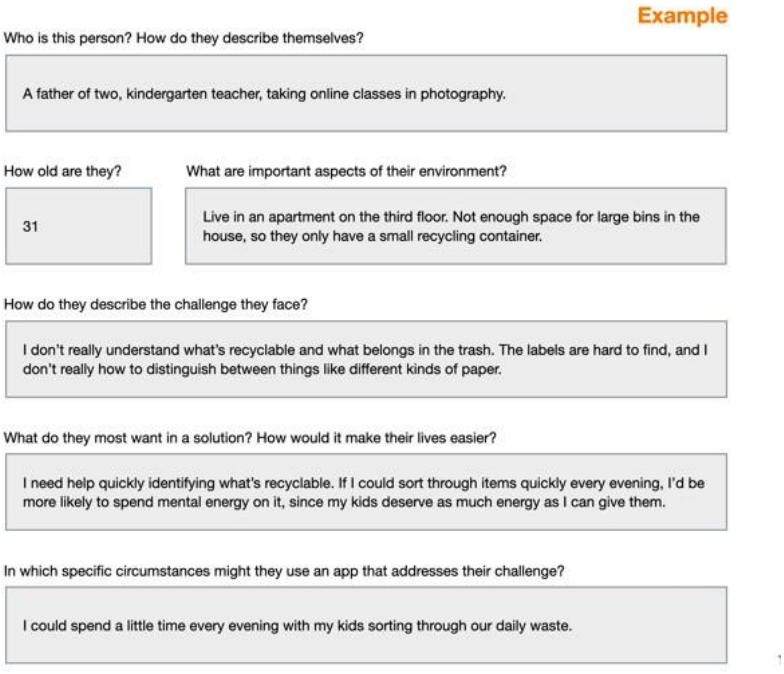
11. Imagine that partway through the day a user has taken 3,467 steps out of the 10,000 step goal. This means he/she is 34.67% of the way to his/her goal. Assign 34.67 to `percentCompleted`. Does the code compile? Go back and explicity assign a type to `percentCompleted` that will allow the code to compile.





Application design workbook

1. Explore your users



2. Consider diversity

Consider Diversity

Identify things about your users you may have overlooked.

A user's identity and circumstances will have a huge impact on how they'll experience and use an app. Summarize all your users with these different aspects in mind.

Everyone has biases that affect the way they perceive the world. Compensate for your biases so that they don't creep into your app's design.

Did you identify something that you didn't consider when imagining your audience? For example, were all your users of similar age? Consider going back to the earlier exercises with your new insights in mind.

Define | Discover

Ages

Above 18

Genders

All

Languages

English, Hindi and Marathi

Disabilities

NA

Cultures

NA

Economic circumstances

Living situations

3. Summarize your audience

Summarize Your Audience

Summarize your findings about individual users. Refer to your earlier research and use it to draw some conclusions.

Define | Discover

Example

What's the most important concern in a solution?

Understanding the percentage of trash vs. recycling.

The age range of the users is:

15 to 30.

Our app will be opened when...

Throwing things in the trash or recycling.

Our app will be used in this environment:

Inside with connection to Wi-Fi or cell reception.

Our environment will have these limitations:

User may have their hands full.

When designing our app, we need to consider:

Users might not know what qualifies as a recyclable.

4. Analyze causes

Analyze Causes

Dig deeper into the issues you've observed and find the core problem. Then consider how your app could solve it.

Asking why something happens will help you discover hidden causes behind what you observe directly. The deeper you dig, the closer you'll get to the core motivating need for your solution.

Create as many copies of the following template as you need to describe the problems you've identified in your research.

Define | Analyze

Example

Users are having this problem:

1. They'd like to begin recycling but struggle to hold themselves accountable.

This happens because:

2. It's easier to throw everything into one bin.

This is because:

3. They struggle to differentiate between trash and recycling.

Which is because:

4. Recycling seems complicated and hard to remember.

And the root cause is this core problem:

5. People are rarely taught how to recycle.

We can solve this issue in our app by:

Educating people on what qualifies as recycling and gamifying the experience so they can hold themselves accountable with their peers.

5. Research Competitors

Research Competitors

Find and describe apps that relate to the problem you've identified.

Discover what people are currently using to solve the problem. Search the App Store for similar apps to find out what users enjoy or dislike about their solution. This will give you insight into what your app will be competing with.

Define | Analyze

This app is interesting because:	I like/dislike this app because:
Easy to use	Only shows expenses
Provides daily expense tracker	No visual charts available
GUI	Lots of advertisements
Recommendation for daily input	Unable to provide expense and saving for certain period
Easy to create balance sheet at the end of month	Unable to attach proof

 # Explore your users

Who is the person? How do they describe themselves?
An ordinary person who likes listening to music.

How old are they?
20

What are important aspects of their environment?
Likes listening to music but cannot find a proper playlists of songs to listen to according to their mood and language preference.

How do they describe the challenge they face?
I can't find a playlist that I would like to listen to, depending upon my mood.

What do they most want in the solution? How would it make their lives easier?
I need multiple options of a person's mood and after clicking the option I would be redirected to a playlist containing all the songs suitable to my mood and according to my language preference.

In which specific circumstances might they use an app that addresses their challenge?
After an exhausting day of work I can listen to a playlist of songs depending upon my mood.

 # Consider diversity

1. **Ages:** 18 and above.
2. **Languages:** No language barrier
3. **Disabilities:** NA
4. **Cultures:** NA
5. **Economic circumstances:**
6. **Living situations:**

 # Summarize your audience

What's the most important concern in a solution?
Understanding the taste of the user in music according to his/her mood.

The age range of the users is:
18 and above

Our app will be opened when..
User wants to play a particular playlist of songs that can help him/her to handle their mood/emotions.

Our app will be used in this environment:
Everywhere with connection to Wi-Fi or Internet data.

Our environment will have these limitations:
NA

When designing our app we need to consider:
Users might not know about the playlist that can handle their current emotions.

 # Analyse causes

Users are having this problem:
They'd like to listen to a playlist of songs depending on their mood but cannot find one**.**

This happens because:
It's difficult to find a playlist related to their current mood.

This is because
They struggle to find songs related to their current mood.

Which is because
Sorting songs in a playlist of a particular genre or feeling or emotion seems time consuming.

And the root cause is this core problem:
People need quick access to a playlist which can support their mood.

We can solve this issue in our app by:
Creating multiple options of user's emotions for example: sad, happy, angry, lonely, etc. Which can give quick access to a playlist of user's choice of mood.