CS 51 Computer Science Principles

APCSP Module 3: Data, Internet, Computer and Programming

AP CSP Review

AP CREATE TASK GUIDELINES (EFFECTIVE 2023)

DR. ERIC CHOU

IEEE SENIOR MEMBER

APCSP Exam Format (Effective 2024)

Section	Question Type/Component	Number of Questions	Exam Weighting	Timing
I	Multiple-choice questions	**70**	**70%**	**120 minutes End-of-course AP Exam**
	Single-select	57		
	Single-select with reading passage about a computing innovation	5		
	Multi-select	8		
II	Create Performance Task	***See Below***	**30%**	***See Below***
	Program code, video, and Personalized Project Reference			At least 9 hours in class
	Written response related to the Create performance task	4		60 minutes End-of-course AP Exam

Big Ideas	Exam Weighting
Big Idea 1: Creative Development	**10–13%**
Big Idea 2: Data	**17–22%**
Big Idea 3: Algorithms and Programming	**30–35%**
Big Idea 4: Computer Systems and Networks	**11–15%**
Big Idea 5: Impact of Computing	**21–26%**

Multiple Choice Questions by Units

APCSP Portfolio Submission and Written Response Exam

LECTURE 1

AP Students

Home | Getting Started | Courses & Exams | Taking Exams

Digital Portfolios

Some AP courses include a portfolio component that needs to be submitted digitally to be scored by the AP Program. You'll upload and submit your work through the AP Digital Portfolio, a secure web application.

Submit AP Computer Science Principles Work in the AP Digital Portfolio

- **30% of your final AP Exam score** in this course comes from your score on the Create performance task. For this task, you'll develop a computer program, record a video of your program running, and develop a Personalized Project Reference containing screen captures of your list and procedures.
- You'll submit these components to College Board through the AP Digital Portfolio. Be sure to submit them as final by no later than **April 30, 2024, 11:59 p.m. ET** so they can be scored.

Instructions

1. Access the AP Digital Portfolio

- Go to the AP Digital Portfolio and sign in using your College Board login information. This is the same login information you use to sign in to My AP.

Instructions

2. Go to your class

- From the dashboard, navigate to your AP Computer Science Principles class.

Instructions

3. Navigate to one of the Create performance task components

From the left navigation menu, choose one of the Create performance task components.

- Program Code
- Video
- Personalized Project Reference

Your Code Listing:

```
def square(x):
    return x * x

def main():
    print(square(3))

if __name__ == “__main__”:
    main()
```

Your Video:

Your Written Responses:

CollegeBoard AP

Create – Applications From Ideas
Written Response Submission Template

Please see Assessment Overview and Performance Task Directions for Student for the task directions and recommended word counts.

Program Purpose and Development

2a)

Blackjack is an online game of 21 I made in Python. The purpose of this program is to help users devlop critical thinking skills and practice concentration through an intense game.The program fulfills this purpose by dealing out a set of cards and promting the user to make a deciosn based on the balue of their hand, promting them to think ahead and then allowing them to review their past actions and keep score of the game. This video illustrates the above features y show one full game of 21 and how afte the user is given a hand they mist decide to hit or stand and at the end the game decides who wins based on the basic rules of 21. Addiotnally, the video shows how the progrm finnaly prints out who won the round and the total score while also allowing the user to review their deicsions.

New for Written Response

1. The written response section will be taken in the new extra hour of the 3-hour exam

2. The personalized project reference listing still need to be submitted.

3. Student's code need to be done to include all the requirements that college board posted. Then, students will have access to the submitted code and write down their written response in the extra exam hour.

Instructions

4. Follow the directions on the screen for each component.

For the "Program Code" component, you'll upload a file.

- Click **Upload New** and select the correct file from your computer.
  Remember: Until you submit the file as final, you can go back and replace it with an updated file.

For the "Video" component, you'll also upload a file.

- Click **Upload New** and select the correct file from your computer.
  Remember: Until you submit the file as final, you can go back and replace it with an updated file.

For the "Personalized Project Reference" component, you'll insert images of segments of your program code related to your list and procedure into text fields.

Instructions

5. Do a final review of your work.

Once you submit your file as final, you won't be able to make any changes. We recommend that you review your work before you submit it as final.

NOTE: All three components must be submitted as final. Only components that have been submitted as final will be sent for scoring. The Personalized Project Reference that you'll receive on the end-of-course exam will be the one that you submitted as final in the AP Digital Portfolio. If you fail to submit your Personalized Project Reference by the deadline, you will not have access to this resource.

Instructions

6. Submit your work as final.

Click Submit Final and go through the final submission process, which includes attesting to the originality of your work. Check the boxes to affirm that you have followed the Performance Task guidelines and have not plagiarized your submission. Each of the three components of the Create performance task must be submitted as final by the deadline to be sent for scoring.

NOTE: All three components must be submitted as final, including attesting to the originality of your work for each component. Only components that have been submitted as final will be sent for scoring. The Personalized Project Reference you'll receive on the end-of-course exam will be the one that you submitted as final in the AP Digital Portfolio. If you fail to submit your Personalized Project Reference by the deadline, you will not have access to this resource.

Design Framework and Programming Languages

LECTURE 2


Scratch
File
Edit
Tips
Sign in to save
Sign in
Full 16 Frame Scratch Cat Walk Cycle
by griffpatch
v311
Scripts
Costumes
Sounds
Remix
See project page
Motion
Looks
Sound
Pen
Data
Events
Control
Sensing
Operators
More Blocks
when clicked
when space key pressed
when this sprite clicked
when backdrop switches to backdrop
when loudness > 10
when I receive message1
broadcast message1
broadcast message1 and wait
forever
wait 0.02 secs
next costume
when this sprite clicked
set size to 50 %
go to x: -195 y: -123
clear
switch costume to Walk1
repeat 8
stamp
change x by 55
next costume
set size to 120 %
go to x: 0 y: 55
x: 5
y: 1
x: 240 y: -180
Sprites
New sprite:
Stage
1 backdrop
New backdrop:
Scratch Cat
Tree_3
Glass-Tal...
Glass-Tal...
grass3
grass2
Tree_2

MIT Scratch

Advantage:

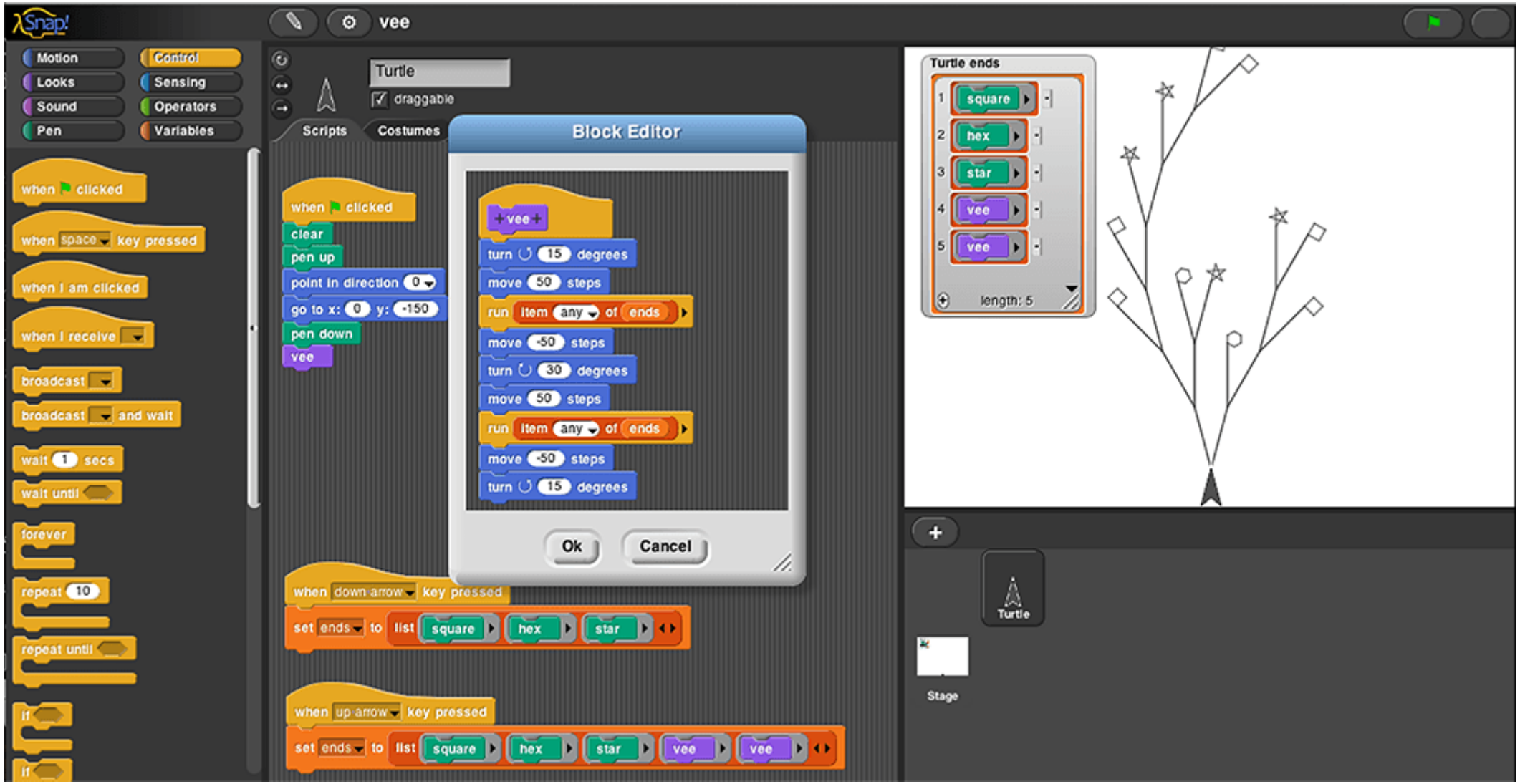
a) Easy to start

b) Shorter learning curve

Disadvantage:

a) Code Listing by picture

b) Not easy to show data structure (list) and function


Snap!
vee
Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
when clicked
when space key pressed
when I am clicked
when I receive
broadcast
broadcast and wait
wait 1 secs
wait until
forever
repeat 10
repeat until
if
if
Turtle
draggable
Scripts
Costumes
when clicked
clear
pen up
point in direction 0
go to x: 0 y: -150
pen down
vee
Block Editor
+vee+
turn 15 degrees
move 50 steps
run item any of ends
move -50 steps
turn 30 degrees
move 50 steps
run item any of ends
move -50 steps
turn 15 degrees
Ok
Cancel
when down-arrow key pressed
set ends to list square hex star
when up-arrow key pressed
set ends to list square hex star vee vee
Turtle ends
1 square
2 hex
3 star
4 vee
5 vee
length: 5
Turtle
Stage

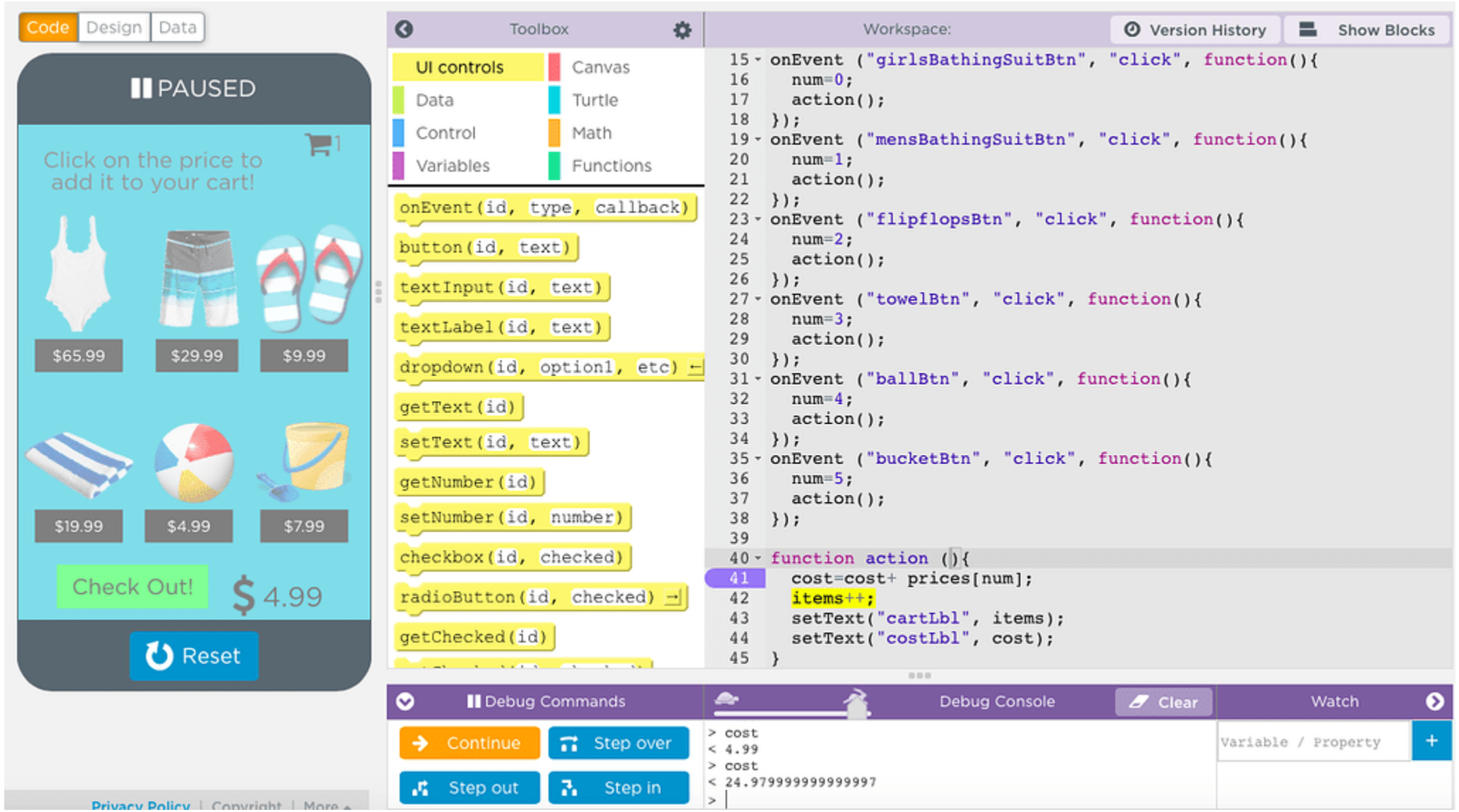
UC Berkeley Snap!

Advantage:

a) Easy to start
b) Shorter learning curve

Disadvantage:

a) Code Listing by picture
b) Not easy to show data structure (list) and function


Code
Design
Data
PAUSED
Click on the price to add it to your cart!
$65.99
$29.99
$9.99
$19.99
$4.99
$7.99
Check Out!
$ 4.99
Reset
Toolbox
UI controls
Canvas
Data
Turtle
Control
Math
Variables
Functions
onEvent(id, type, callback)
button(id, text)
textInput(id, text)
textLabel(id, text)
dropdown(id, option1, etc)
getText(id)
setText(id, text)
getNumber(id)
setNumber(id, number)
checkbox(id, checked)
radioButton(id, checked)
getChecked(id)
Workspace:
Version History
Show Blocks
15 onEvent ("girlsBathingSuitBtn", "click", function(){
16 num=0;
17 action();
18 });
19 onEvent ("mensBathingSuitBtn", "click", function(){
20 num=1;
21 action();
22 });
23 onEvent ("flipflopsBtn", "click", function(){
24 num=2;
25 action();
26 });
27 onEvent ("towelBtn", "click", function(){
28 num=3;
29 action();
30 });
31 onEvent ("ballBtn", "click", function(){
32 num=4;
33 action();
34 });
35 onEvent ("bucketBtn", "click", function(){
36 num=5;
37 action();
38 });
39
40 function action (){
41 cost=cost+ prices[num];
42 items++;
43 setText("cartLbl", items);
44 setText("costLbl", cost);
45 }
Debug Commands
Continue
Step over
Step out
Step in
Debug Console
Clear
> cost
< 4.99
> cost
< 24.979999999999997
>
Watch
Variable / Property
+

Code.org App Lab

Advantage:

a) Easy to start
b) Shorter learning curve
c) Good code listing (block mode/JavaScript mode switch very easy)

Disadvantage:

a) None for the exam purpose
b) Small project size.


Python Pygame
www.educba.com

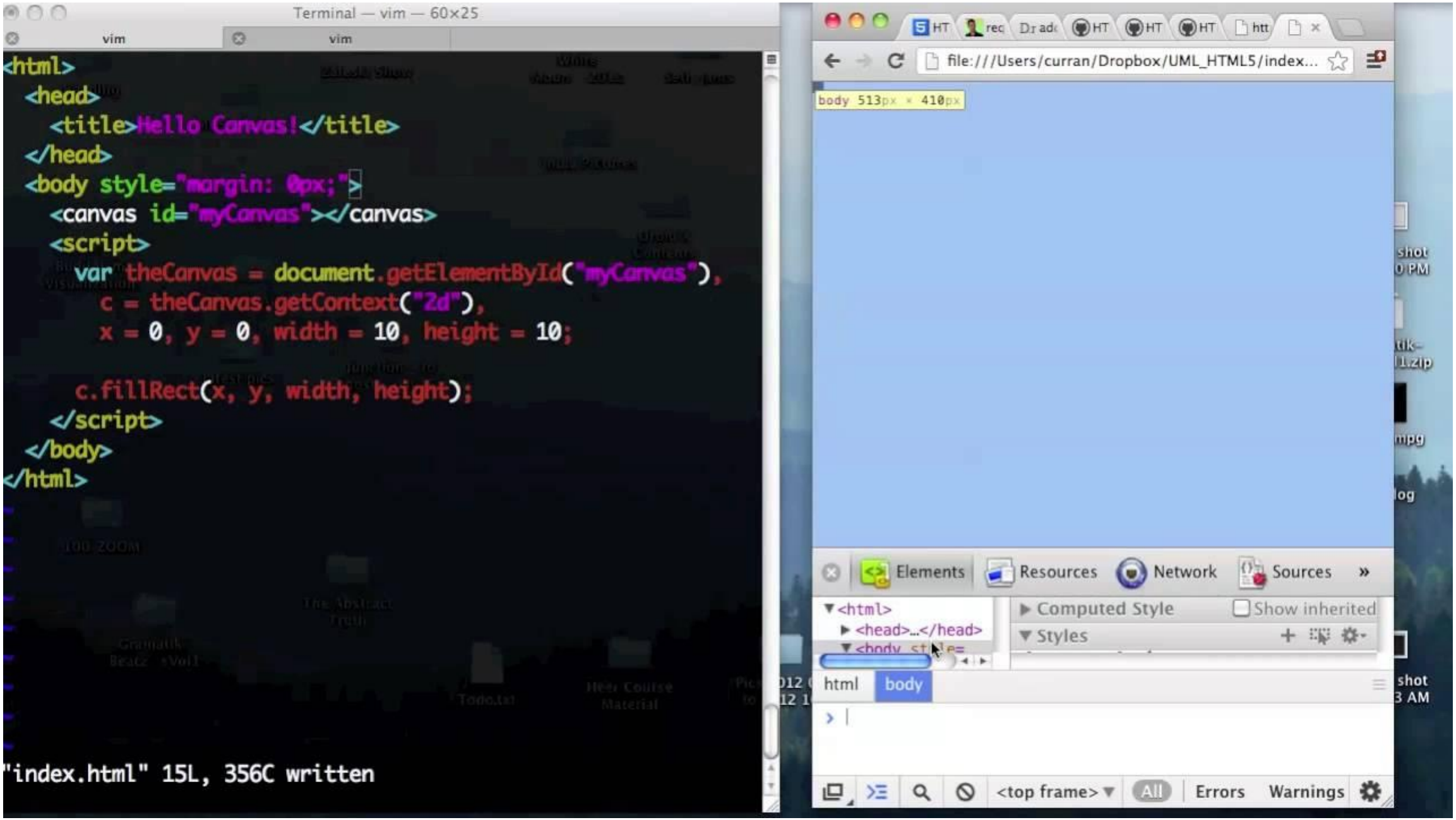
Python PyGame

Advantage:

a) Complete GUI Design
b) Good for code listing
c) Easy to show data structure and function
d) Large project size possible

Disadvantage:

a) Need to have some learning curve
b) May overkill the purpose of the APCSP Exam


Terminal — vim — 60×25
vim
vim
<html>
<head>
<title>Hello Canvas!</title>
</head>
<body style="margin: 0px;">
<canvas id="myCanvas"></canvas>
<script>
var theCanvas = document.getElementById("myCanvas"),
c = theCanvas.getContext("2d"),
x = 0, y = 0, width = 10, height = 10;
c.fillRect(x, y, width, height);
</script>
</body>
</html>
"index.html" 15L, 356C written
file:///Users/curran/Dropbox/UML_HTML5/index...
body 513px × 410px
Elements
Resources
Network
Sources
<html>
<head>…</head>
Computed Style
Show inherited
Styles
html
body
<top frame>
All
Errors
Warnings

JavaScript Canvas

Advantage:

a) Complete GUI Design
b) Good for code listing
c) Easy to show data structure and function
d) Large project size possible

Disadvantage:

a) Need to have some learning curve

Organizing Your Projects

LECTURE 3

Create Task

- 12 Hours in class (home time also OK) to create program code, video, and project's written response (reference)
- Written response now need to be given in the Final exam. All grading is based on the final exam turnout and the submitted digital portfolio.

Section II: Create Performance Task: Written Responses

The Create performance task section of the end-of-course exam consists of four prompts that require students to write responses that demonstrate understanding of their personal Create performance task. The following are sample prompts for each of the four categories – but the specific prompts will vary across the different versions of the exam. Students will have access to their student-authored Personalized Project Reference, as long as it was submitted as final via the AP Digital Portfolio, when responding to these prompts.

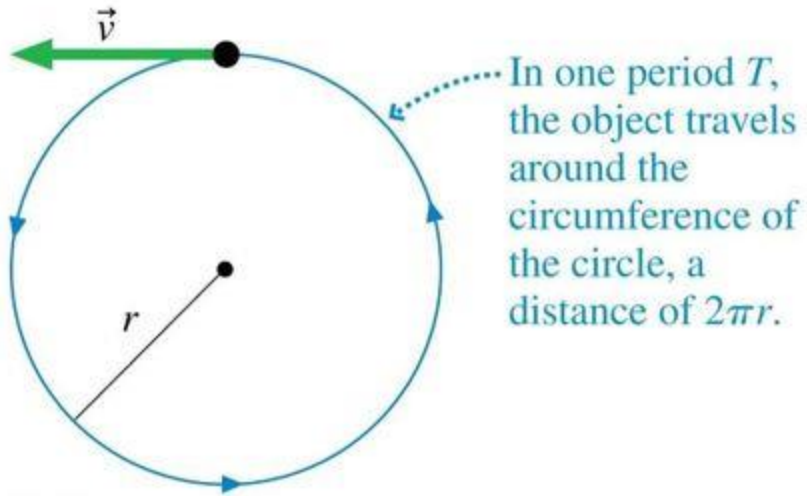
Prompt Category	Sample Prompt
Program Design, Function, and Purpose	Identify an expected user of your program. Describe one way your program's design meets the needs of this user.
Algorithm Development	Consider the first iteration statement included in the Procedure section of your Personalized Project Reference. Identify the number of times the body of your iteration statement will execute. Describe a condition or error that would cause your iteration statement to not terminate and cause an infinite loop. If no such condition or error exists, explain how the loop could be modified to cause an infinite loop.
Errors and Testing	Consider the procedure included in part (i) of the Procedure section of your Personalized Project Reference. Describe a change to your procedure that will result in a run-time error. Explain why this change will result in a run-time error.
Data and Procedural Abstraction	Suppose you are provided with a procedure called `isEqual (value1, value2)`. The procedure returns `true` if the two parameters `value1` and `value2` are equal in value and returns `false` otherwise. Using the list you identified in the List section of your Personalized Project Reference, explain in detailed steps an algorithm that uses `isEqual` to count the number of times a certain value appears in your list. Your explanation must be detailed enough for someone else to write the program code.

Program's Design, Purpose and Function

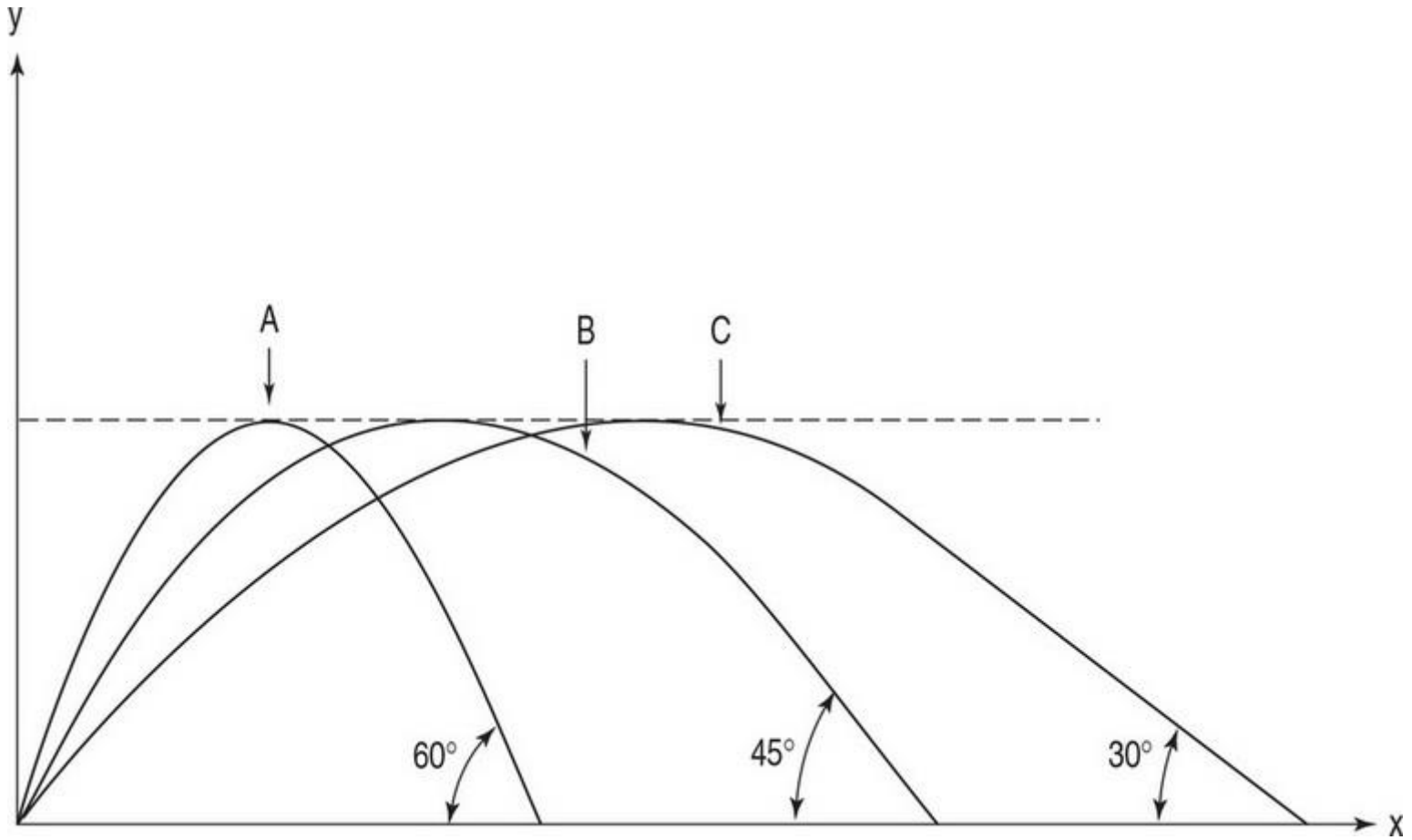
- Identify an expected user of your program. Describe one way your program's design meets the needs of this user.
- No Game.
- Educational more than entertainment.
- AP Physics C, AP Chemistry, AP Biology

Uniform Circular Motion

- Uniform circular motion describes an object that travels along a circular trajectory at a constant speed.
- The velocity of the object is always tangent to the trajectory of the object.
- The magnitude of the velocity for uniform circular motion is defined using the circumference of the path and the period.
- Period (T) is the amount of time required to travel once around the circle.



$$v = \frac{2\pi r}{T}$$



An Ideal Gas (in a box)

PV = nRT

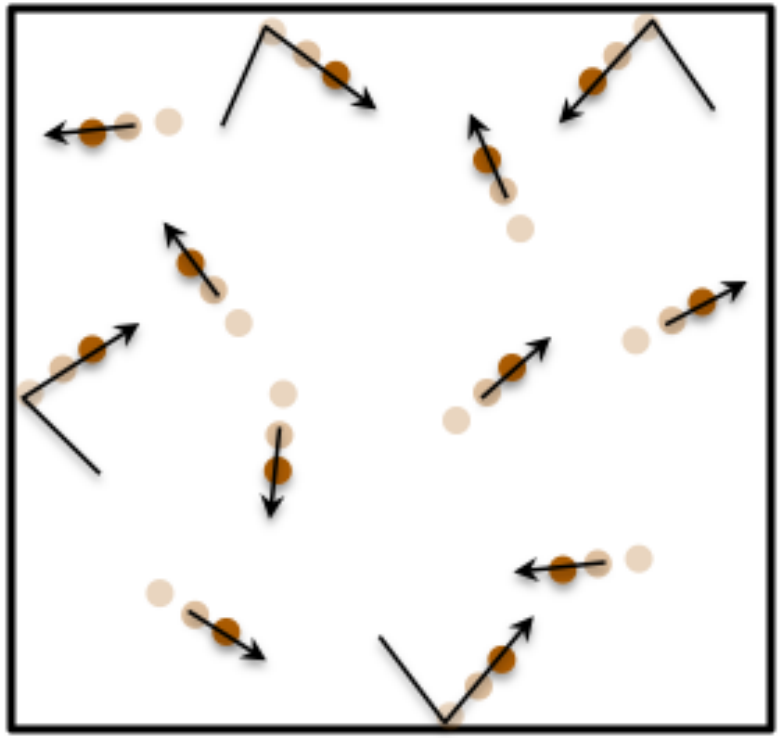
P = pressure

V = volume

n = number

T = temperature

(and R is just a number)



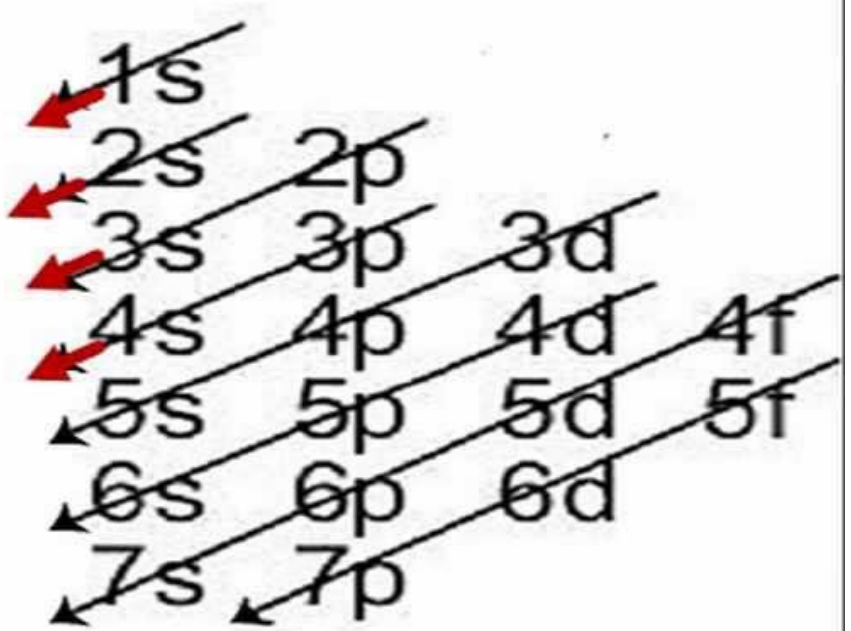
T ↑: they go faster

P: is caused by them hitting the walls

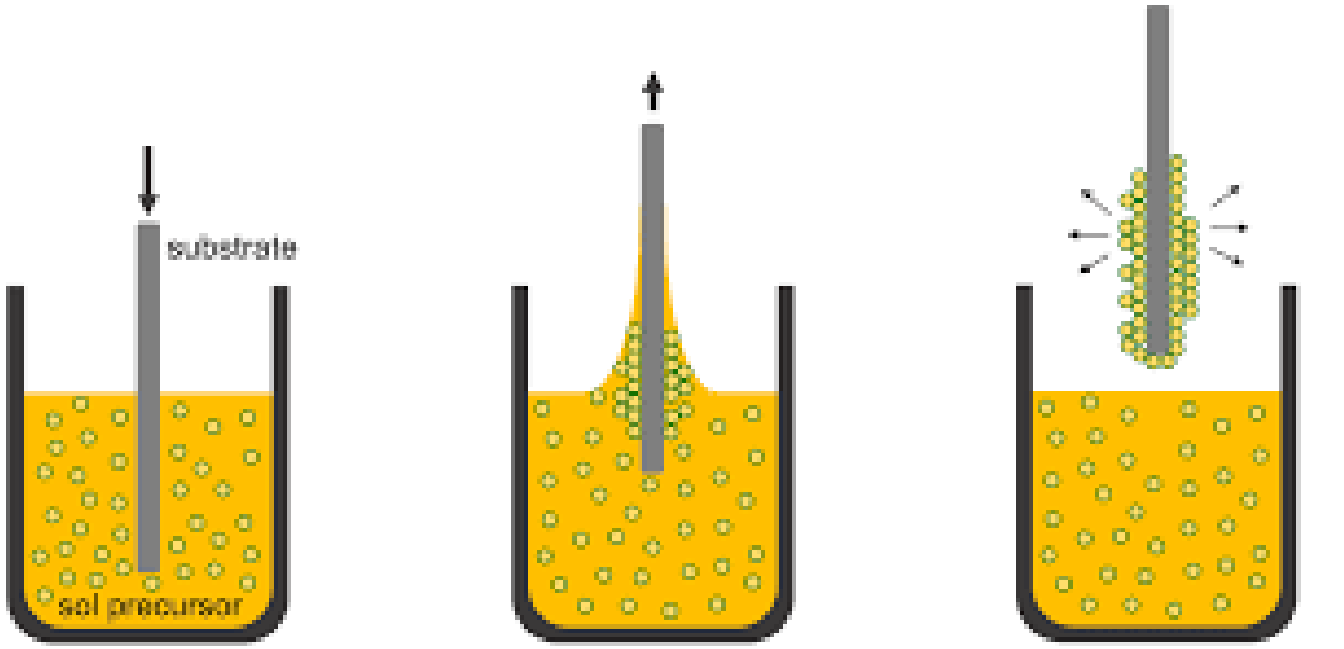
Electron Configuration Chart

s holds up to **2** **p** holds up to **6** **d** holds up to **10**

19
K
Potassium
39.10



$$1s^2\ 2s^2\ 2p^6\ 3s^2\ 3p^6\ 4s^1$$



a) dipping b) withdrawal c) evaporation

Algorithm Development

- Consider the first iteration statement included in the Procedure section of your Personalized Project Reference. Identify the number of times the body of your iteration statement will execute. Describe a condition or error that would cause your iteration statement to not terminate and cause an infinite loop. If no such condition or error exists, explain how the loop could be modified to cause an infinite loop.

Checklist

- Functional definition
- Parameter and return value
- Sequential programming
- Selection Programming
- Iterative Programming
- Testing and debugging on the logical branches

Errors and Testing

- Consider the procedure included in part (i) of the Procedure section of your Personalized Project Reference. Describe a change to your procedure that will result in a run-time error. Explain why this change will result in a run-time error.

Data and Procedural Abstraction

Suppose you are provided with a procedure called **isEqual (value1, value2).** The procedure returns true if the two parameters value1 and value2 are equal in value and returns false otherwise. Using the list you identified in the List section of your Personalized Project Reference, explain in detailed steps an algorithm that uses isEqual to count the number of times a certain value appears in your list. Your explanation must be detailed enough for someone else to write the program code.

Written Response Problems

LECTURE 3

General Requirements

You will be provided with a minimum of 9 hours of class time to complete and submit the following:

- **Final program code** (created independently or collaboratively)
- **A video that displays the running of your program and demonstrates functionality you developed** (created independently)
- **Code Segments for your Personalized Project Reference** (created independently)

Note: Students in nontraditional classroom environments should consult a school-based AP Coordinator for instructions.

Teamwork is not recommended

Even though college board encourage students to perform in a team format. But this APCSP portfolio has the following nature to prevent good teamwork:

1. Project size too small
2. very hard to differentiate whose credit it is.
3. It is hard to write a report if the whole project is a team effort.
4. It is hard to have every student to be evenly divided among the program development.
5. If your part does not have all ingredients that the college board requires, then, you will be in a big trouble.

Your APCSP Portfolio Score is largely depends on your written response

- Program code, video just for reference.
- The written response is 30% of your final score in the APCSP.
- The written response will be read at the same time that your MCQ is graded.

Submission Requirements

COMPONENT A: PROGRAM CODE (CREATED INDEPENDENTLY OR COLLABORATIVELY)

Submit one PDF file that contains all of your program code (including comments). Include comments or acknowledgments for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

IMPORTANT:

If the programming environment allows you to include comments, this is the preferred way to acknowledge and give credit to another author. However, if the programming environment does not allow you to include comments, you can add them in a document editor when you capture your program code for submission.

Submission Requirements

In your program, you must include student-developed program code that contains the following:

- **Instructions for input from one of the following:**
  - the user (including user actions that trigger events)
  - a device
  - an online data stream
  - a file

Submission Requirements

- ☐ Use of at least one **list** (or other **collection type**) to represent a collection of data that is stored and used to manage program complexity and help fulfill the program's purpose

IMPORTANT:

The data abstraction must make the program easier to develop (alternatives would be more complex) or easier to maintain (future changes to the size of the list would otherwise require significant modifications to the program code).

Submission Requirements

- At least one procedure that contributes to the program's intended purpose, where you have defined:
  - the procedure's name
  - the return type (if necessary)
  - one or more parameters

IMPORTANT:

Implementation of built-in or existing procedures or language structures, such as event handlers or main methods, are not considered student-developed.

Submission Requirements

- ☐ An algorithm that includes sequencing, selection, and iteration that is in the body of the selected procedure
- ☐ Calls to your student-developed procedure
- ☐ Instructions for output (tactile, audible, visual, or textual) based on input and program functionality

Submission Requirements

- ☐ An algorithm that includes sequencing, selection, and iteration that is in the body of the selected procedure
- ☐ Calls to your student-developed procedure
- ☐ Instructions for output (tactile, audible, visual, or textual) based on input and program functionality

Video

LECTURE 4

Submission Requirements

COMPONENT B: VIDEO (CREATED INDEPENDENTLY)

Submit one video file that demonstrates the running of your program as described below. Collaboration is not allowed during the development of your video.

Submission Requirements

Your video must demonstrate your program running, including:

- Input to your program
- At least one aspect of the functionality of your program
- Output produced by your program

Your video may NOT contain:

- Any distinguishing information about yourself
- Voice narration (though text captions are encouraged)

Your video must be:

- Either .webm, .mp4, .wmv, .avi, or .mov format
- No more than 1 minute in length
- No more than 30MB in file size

Personalized Project Reference

LECTURE 5

Submission Requirements

COMPONENT C: PERSONALIZED PROJECT REFERENCE (CREATED INDEPENDENTLY)

To assist in responding to the written response prompts on exam day, submit required portions of your code by capturing and pasting program code segments you developed during the administration of this task.

Screen captures should not be blurry, and text should be at least 10-point font size. Your code segments should not include any comments. These code segments will be made available to you on exam day only if t

Procedure: Capture and paste two program code segments you developed during the administration of this task that contain a student-developed procedure that implements an algorithm used in your program and a call to that procedure.

IMPORTANT:

Built-in or existing procedures and language structures, such as event handlers and main methods, are not considered student-developed.

i. The first program code segment must be a student-developed procedure that:

- ☐ Defines the procedure's name and return type (if necessary)
- ☐ Contains and uses one or more parameters that have an effect on the functionality of the procedure
- ☐ Implements an algorithm that includes sequencing, selection, and iteration

ii. The second program code segment must show where your student-developed procedure is being called in your program.

List: Capture and paste two program code segments you developed during the administration of this task that contain a list (or other collection type) being used to manage complexity in your program.

i. The first program code segment must show how data have been stored in the list.

ii. The second program code segment must show the data in the same list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.

Introduction Video for APCSP Create Task

LECTURE 6

Dr. Wu's Introducing the AP CSP Create Task

https://youtu.be/XD260Ywh5TI?si=SW0k9d7wIlcjUzss

BMI Program A Sample AP Create Task Demo

LECTURE 7

Body Mass Index

The formula for calculating BMI was first invented in 1832 by Belgian mathematician and astronomer Adolphe Quetelet. To calculate it, you divide weight in kilograms by height in metres squared (kg/m2). In adults, BMI is categorised as follows:

BMI	Weight status
Below 18.5	Underweight
18.5-24.9	Normal weight
25.0-29.9	Overweight
30.0-34.9	Obesity class I
35.0-39.9	Obesity class II
Above 40	Obesity class III

CodingDemos