

| Paper coding

Try to fully understand the basic concept before moving on to the next step.

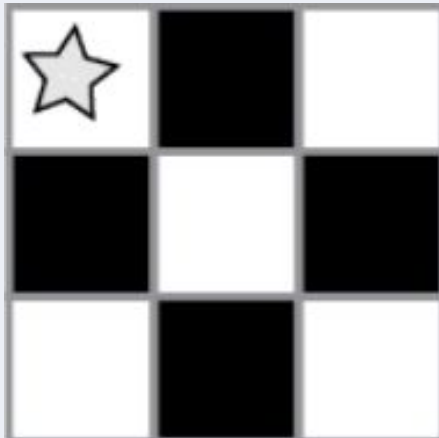
Lack of understanding basic concepts will increase your burden in learning this course, which may make you fail the course.

It may be difficult now, but for successful completion of this course we suggest you to fully understand the concept and move on to the next step.

Q1 Python codes are executed in sequence. Image the 3x3 box below. Starting with the first box, each box is colored according to the command below. One of many possible algorithms is shown below.

▪

Start
Here



<https://code.org/curriculum/course2/1/Teacher>

Move Right,
Fill-In Square,
Move Right,
Move Down
Fill-In Square,
Move Left,
Move Left,
Fill-In Square
Move Down,
Move Right,
Fill-In Square,
Move Right



Write the entire code and the expected output results in the note.

Q2 Starting with the first box at the top left, move one space at a time, and fill in the boxes as below.
Think about the more efficient way to execute.

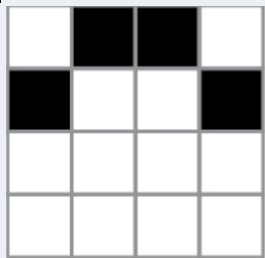


Image
1

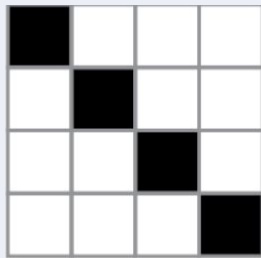


Image 2

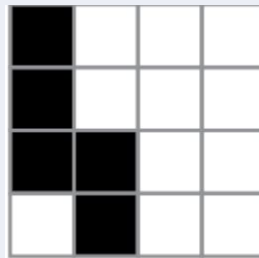


Image 3

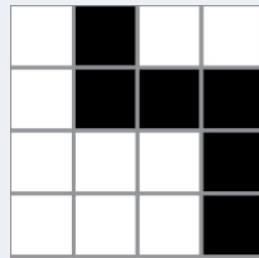


Image 4

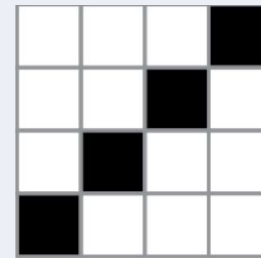


Image 5

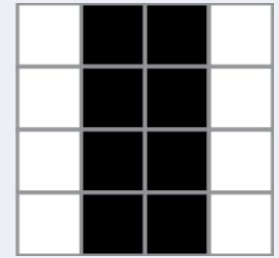
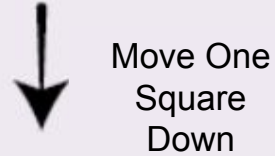
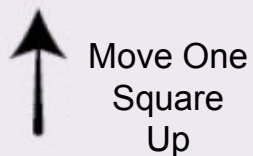



Image 6



<https://code.org/curriculum/course2/1/Teacher>

 Write the entire code and the expected output results in the note.



Pair Programming Practice

| Guideline, mechanisms & contingency plan

Preparing pair programming involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns “driving the mouse.” Effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, it is important to make it clear that the active student will not be punished because the pairing did not work well.

| Pairing similar, not necessarily equal, abilities as partners

Pair programming can be effective when students of similar, though not necessarily equal, abilities are paired as partners. Pairing mismatched students often can lead to unbalanced participation. Teachers must emphasize that pair programming is not a “divide-and-conquer” strategy, but rather a true collaborative effort in every endeavor for the entire project. Teachers should avoid pairing very weak students with very strong students.

| Motivate students by offering extra incentives

Offering extra incentives can help motivate students to pair, especially with advanced students. Some teachers have found it helpful to require students to pair for only one or two assignments.



Pair Programming Practice

| Prevent collaboration cheating

The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

| Collaborative learning environment

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

Q1 Code seven lines of print() function to produce a result like below.

```
      *  
     ***  
    *****  
   *********  
  *****  
   *****  
    ***  
     *
```

Samsung SW Qualification Test sample question

| Paper coding

Try to fully understand the basic concept before moving on to the next step.

Lack of understanding basic concepts will increase your burden in learning this course, which may make you fail the course.

It may be difficult now, but for successful completion of this course we suggest you to fully understand the concept and move on to the next step.



Pair Programming Practice

| Guideline, mechanisms & contingency plan

Preparing pair programming involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns “driving the mouse.” Effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, it is important to make it clear that the active student will not be punished because the pairing did not work well.

| Pairing similar, not necessarily equal, abilities as partners

Pair programming can be effective when students of similar, though not necessarily equal, abilities are paired as partners. Pairing mismatched students often can lead to unbalanced participation. Teachers must emphasize that pair programming is not a “divide-and-conquer” strategy, but rather a true collaborative effort in every endeavor for the entire project. Teachers should avoid pairing very weak students with very strong students.

| Motivate students by offering extra incentives

Offering extra incentives can help motivate students to pair, especially with advanced students. Some teachers have found it helpful to require students to pair for only one or two assignments.



Pair Programming Practice

I Prevent collaboration cheating

The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

I Collaborative learning environment

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

Q1

Express the way of finding the hotel in pseudocode using the pseudocode commands given below.
Write everything else in everyday language except the commands below.
Pay attention to indentation when using commands.

**“Go straight west at this crossroads. You will see four buildings.
When you see the post office, turn right across the post office.
Go straight, and when you see an Italian restaurant, the third building is the hotel you are looking for.”**

Commands to use

- Input : read, obtain, get
- output: print, show, display
- Calculation: compute, calculate, determine
- Iteration: for, while
- Decision-making according to conditions: if-then-else
- Conditional iteration: repeat-until
- Correct, wrong: true, false



Write the entire code and the expected output results in the note.

| Pair programming



Pair Programming Practice

| Guideline, mechanisms & contingency plan

Preparing pair programming involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns “driving the mouse.” Effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, it is important to make it clear that the active student will not be punished because the pairing did not work well.

| Pairing similar, not necessarily equal, abilities as partners

Pair programming can be effective when students of similar, though not necessarily equal, abilities are paired as partners. Pairing mismatched students often can lead to unbalanced participation. Teachers must emphasize that pair programming is not a “divide-and-conquer” strategy, but rather a true collaborative effort in every endeavor for the entire project. Teachers should avoid pairing very weak students with very strong students.

| Motivate students by offering extra incentives

Offering extra incentives can help motivate students to pair, especially with advanced students. Some teachers have found it helpful to require students to pair for only one or two assignments.



Pair Programming Practice

| Prevent collaboration cheating

The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

| Collaborative learning environment

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

Q1. Write a flowchart factorizes 18, 39, 63, 126, 792

| Paper coding

Try to fully understand the basic concept before moving on to the next step.

Lack of understanding basic concepts will increase your burden in learning this course, which may make you fail the course.

It may be difficult now, but for successful completion of this course we suggest you to fully understand the concept and move on to the next step.

Q1 Use the int function to output the number 100 by adding the letter "50" and the number 50.
▪ Also, use the str function to output the result of the addition "5050".

Condition for Execution	The first line outputs the number 100, and the second line outputs the letter "5050".
Time	5minutes



Write the entire code and the expected output results in the note.

Q2. One letter '1' and three '0' are given. Use these two to make the number 1000. Here, only addition operations can be used between strings, and the int() function can be used only once.

Condition for Execution	Output the number 1000.
Time	5minutes



Write the entire code and the expected output results in the note.

| Pair programming



Pair Programming Practice

| Guideline, mechanisms & contingency plan

Preparing pair programming involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns “driving the mouse.” Effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, it is important to make it clear that the active student will not be punished because the pairing did not work well.

| Pairing similar, not necessarily equal, abilities as partners

Pair programming can be effective when students of similar, though not necessarily equal, abilities are paired as partners. Pairing mismatched students often can lead to unbalanced participation. Teachers must emphasize that pair programming is not a “divide-and-conquer” strategy, but rather a true collaborative effort in every endeavor for the entire project. Teachers should avoid pairing very weak students with very strong students.

| Motivate students by offering extra incentives

Offering extra incentives can help motivate students to pair, especially with advanced students. Some teachers have found it helpful to require students to pair for only one or two assignments.



Pair Programming Practice

| Prevent collaboration cheating

The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

| Collaborative learning environment

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

Q1 $n!$ is defined as $n * (n-1) * (n-2) * \dots * 2 * 1$.
Find 5! and 10! using integer and * operator and print them as following.

Output Example

Calculate the factorial.
5! = 120

Calculate the factorial.
10! = 3628800

| Paper coding

Try to fully understand the basic concept before moving on to the next step.

Lack of understanding basic concepts will increase your burden in learning this course, which may make you fail the course.

It may be difficult now, but for successful completion of this course we suggest you to fully understand the concept and move on to the next step.

Q1 Assign the values 30, 60 to the variables width and height, respectively.
Write a program that uses these two variables to find the area of the rectangle as shown below.

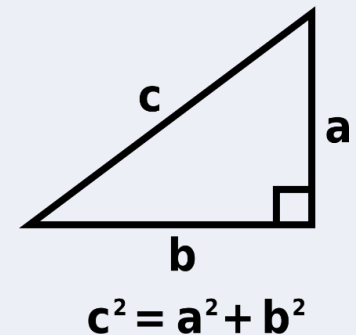
Conditions for Execution	"Print "Area of Rectangle : 1800."
Time	5 Minutes




Write the entire code and the expected output results in the note.

Q2. The Pythagorean Theorem states that the square of the hypotenuse c for any right triangle is equal to the square of the base a plus the square of the height b . Write code that calculates the length of the hypotenuse by receiving the base and height as integers.

Conditions for Execution	Enter the bottom of the first line and the height of the second line. In the third line, print out the result of calculating the length of the hypotenuse.
Time	5 Minutes
The Pythagorean Theorem	$c^2 = a^2 + b^2$



 Write the entire code and the expected output results in the note.

| Pair programming



Pair Programming Practice

| Guideline, mechanisms & contingency plan

Preparing pair programming involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns “driving the mouse.” Effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, it is important to make it clear that the active student will not be punished because the pairing did not work well.

| Pairing similar, not necessarily equal, abilities as partners

Pair programming can be effective when students of similar, though not necessarily equal, abilities are paired as partners. Pairing mismatched students often can lead to unbalanced participation. Teachers must emphasize that pair programming is not a “divide-and-conquer” strategy, but rather a true collaborative effort in every endeavor for the entire project. Teachers should avoid pairing very weak students with very strong students.

| Motivate students by offering extra incentives

Offering extra incentives can help motivate students to pair, especially with advanced students. Some teachers have found it helpful to require students to pair for only one or two assignments.



Pair Programming Practice

I Prevent collaboration cheating

The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

I Collaborative learning environment

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

Q1 Receive a radius value from the user and print the circumference and area of a circle with this radius that follows. Use the variable $PI = 3.141592$ to obtain this value.

**Example
Result**

Enter the radius of a circle: 11

Circumference of a circle= 69.115024, Area of a circle= 280.132632

- Circumference of a circle: $2 * \text{radius} * PI$
- Area(Surface) of a circle: $PI * \text{radius} * \text{radius}$
- Study the area and circumference of a circle before moving forward.

Q2. Write a program that displays the squared values from 2 to 6 in a table as the following. As shown below, a can be increased from 2 to 6, and n has a value of 2. Input the actual value for the part corresponding to $a^{**}n$, so that the output result of the equation is $2^{**}2$.

Example Result	a	n	a ** n
	2	2	4
	3	2	9
	4	2	16
	5	2	25
	6	2	36

| Paper coding

Try to fully understand the basic concept before moving on to the next step.

Lack of understanding basic concepts will increase your burden in learning this course, which may make you fail the course.

It may be difficult now, but for successful completion of this course we suggest you to fully understand the concept and move on to the next step.

Q1. Write code that receives the value n through the user's keyboard input.
Return True if the given integer n is odd and return False if the integer is even.
For cases where n is 20 and 21, print the following.

Conditions for Execution	Enter an Integer : 20 Is the integer odd?: : False Enter an Integer : 21 Is the integer odd?: True
Time	5 Minutes



Write the entire code and the expected output results in the note.

Q2. Write code that takes the user's input and determines whether the integer value n is an even number within the range of 0 to 100 or not. The result of execution should look as follows:

Conditions for Execution	Enter an integer: 12 Is the input an even integer between 0 and 100?
Time	False 5 Minutes



Write the entire code and the expected output results in the note.

| Pair programming



Pair Programming Practice

| Guideline, mechanisms & contingency plan

Preparing pair programming involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns “driving the mouse.” Effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, it is important to make it clear that the active student will not be punished because the pairing did not work well.

| Pairing similar, not necessarily equal, abilities as partners

Pair programming can be effective when students of similar, though not necessarily equal, abilities are paired as partners. Pairing mismatched students often can lead to unbalanced participation. Teachers must emphasize that pair programming is not a “divide-and-conquer” strategy, but rather a true collaborative effort in every endeavor for the entire project. Teachers should avoid pairing very weak students with very strong students.

| Motivate students by offering extra incentives

Offering extra incentives can help motivate students to pair, especially with advanced students. Some teachers have found it helpful to require students to pair for only one or two assignments.



Pair Programming Practice

I **Prevent collaboration cheating**

The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

I **Collaborative learning environment**

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

Q1. Receive a 3-digit integer from the user. If the hundredth digit of the integer `n` is 3, return `True`. If not, return `False`.

Example Result	Enter a 3-digit integer: 321 True
---------------------------	--

- ▶ Hint: You must be familiar with the `//` operator. If not, please go back and review it.

Q2. Receive one integer. If the integer is a multiple of 5, return True. If not, return False.

Example Result	Enter an integer: 125 True
---------------------------	-------------------------------

| Paper coding

Try to fully understand the basic concept before moving on to the next step.

Lack of understanding basic concepts will increase your burden in learning this course, which may make you fail the course.

It may be difficult now, but for successful completion of this course we suggest you to fully understand the concept and move on to the next step.

Q1. If a game user's game_score is over 1000 points, print 'You are a master.'

**Example
Output**

```
Enter game score : 1500  
game_score = 1500  
You are a master.
```

or

```
Enter game score : 100  
game_score = 100
```

Time

5min



Write the entire code and the expected output results in the note.

Q2 Write a program that receives any integer x between -100 and 100 and 1) prints x on the screen, and 2) prints "...is a natural number" if x is an integer greater than zero. Otherwise, let it simply print x as in x = -10.

**Example
Output**

Enter integer:
x = 50
50 is a natural number.

or

Enter integer:
x = -10

Time

5min



Write the entire code and the expected output results in the note.

| Pair programming



Pair Programming Practice

| Guideline, mechanisms & contingency plan

Preparing pair programming involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns “driving the mouse.” Effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, it is important to make it clear that the active student will not be punished because the pairing did not work well.

| Pairing similar, not necessarily equal, abilities as partners

Pair programming can be effective when students of similar, though not necessarily equal, abilities are paired as partners. Pairing mismatched students often can lead to unbalanced participation. Teachers must emphasize that pair programming is not a “divide-and-conquer” strategy, but rather a true collaborative effort in every endeavor for the entire project. Teachers should avoid pairing very weak students with very strong students.

| Motivate students by offering extra incentives

Offering extra incentives can help motivate students to pair, especially with advanced students. Some teachers have found it helpful to require students to pair for only one or two assignments.



Pair Programming Practice

| Prevent collaboration cheating

The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

| Collaborative learning environment

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

Q1 Write a program that receives age as input and prints "Adult" if age is 20 or over, "Youth" if under 20 and equal to or over 10, and "Kid" if under 10.

Output example

```
Enter age: 16  
Youth
```

```
Enter age: 33  
Adult
```

```
Enter age: 5  
Kid
```

Q2. Write a program that prints "Adult" if the age is equal to or over 20, "Youth" if under 20 and equal to or over 10 and "Kid" if under 10.

Output example

```
Enter age: 20
Enter height in cm: 180
You can enter.
```

| Paper coding

Try to fully understand the basic concept before moving on to the next step.

Lack of understanding basic concepts will increase your burden in learning this course, which may make you fail the course.

It may be difficult now, but for successful completion of this course we suggest you to fully understand the concept and move on to the next step.

Q1 Receive an alphabetical letter from the user and print 'It is a vowel' for a, e, i, o, u, and 'It is a consonant' for any other letters.

**Example
Output**

Enter the alphabet : k
K is a consonant

Time
e

5min



Write the entire code and the expected output results in the note.

Q2 Write the following program which receives two integers a, b as input, determines whether a is a multiple of b and prints the result.

Example Output	Write two integers: 30 3 30 is a multiple of 3
Time	5min



Write the entire code and the expected output results in the note.

| Pair programming



Pair Programming Practice

| Guideline, mechanisms & contingency plan

Preparing pair programming involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns “driving the mouse.” Effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, it is important to make it clear that the active student will not be punished because the pairing did not work well.

| Pairing similar, not necessarily equal, abilities as partners

Pair programming can be effective when students of similar, though not necessarily equal, abilities are paired as partners. Pairing mismatched students often can lead to unbalanced participation. Teachers must emphasize that pair programming is not a “divide-and-conquer” strategy, but rather a true collaborative effort in every endeavor for the entire project. Teachers should avoid pairing very weak students with very strong students.

| Motivate students by offering extra incentives

Offering extra incentives can help motivate students to pair, especially with advanced students. Some teachers have found it helpful to require students to pair for only one or two assignments.



Pair Programming Practice

| Prevent collaboration cheating

The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

| Collaborative learning environment

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

Q1

Write a program that executes addition, subtraction, multiplication, and division. It prints the operation result of two positive integers, based on the desired operation number given as input. If a number else than 1, 2, 3 and 4 is given as input, 'Entered an incorrect number' is printed. To enter two numbers, write one, press enter and write another one.

Output example

```
1) Addition  2) Subtraction  3) Multiplication  4) Division
Enter the desired number of operation : 1
Enter two numbers for operation.
10
20
10 + 20 = 30
```

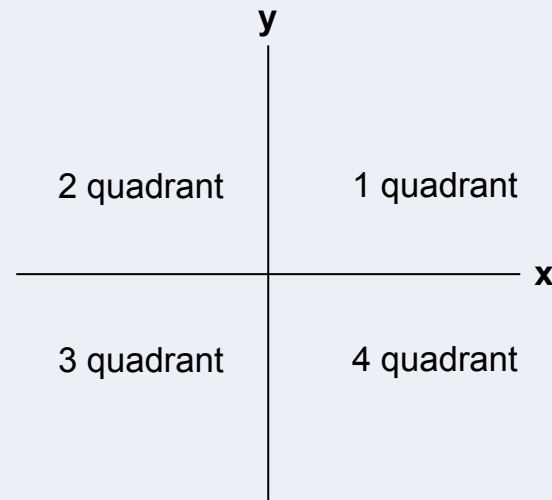
If inserted incorrectly

```
1) Addition  2) Subtraction  3) Multiplication  4) Division
Enter the desired number of operation : 5
Entered an incorrect number.
```

Q2 Write a program that receives a point with x and y coordinates as input, and determines in which quadrant among 1, 2, 3, 4 the point belongs. The position of the quadrant is shown in the following figure.

Output example

```
Enter x,y coordinates: -5 6  
In the second quadrant
```



Q3

Develop a menu ordering program for Yummy Restaurant. Show the following menu to the user and let the user select one. If the given input alphabet is not in the menu, print 'enter the menu again:' and receive another input.

Output example

Welcome to yummy restaurant. Here is the menu.

- Burger(enter b)
- Chicken(enter c)
- Pizza(enter p)

Choose a menu (enter b,c,p) : b

You chose pizza.

- This needs complex conditional expressions. Combine the logical operators and the conditional statements carefully.

| Paper coding

Try to fully understand the basic concept before moving on to the next step.

Lack of understanding basic concepts will increase your burden in learning this course, which may make you fail the course.

It may be difficult now, but for successful completion of this course we suggest you to fully understand the concept and move on to the next step.

Q1 Declaring the list `bts = ['V', 'J-Hope', 'RM', 'Jungkook', 'Jin', 'Jimin', 'Suga']`. Then write a code that prints all items in this list using the for statement.

**Example
Output**

```
V  
J-Hope  
RM  
Jungkook  
Jin  
Jimin  
Suga
```

Time

5min



Write the entire code and the expected output results in the note.

Q2 Use cumulative addition to compute and print the sum of integers from 1 to 100.
(Hint: make the printed value of the range function range from 1 to 100.)

Example Output	Sum of integers from 1 to 100 : 5050
Time	5min



Write the entire code and the expected output results in the note.

Q3 Use the step value of the range function to find the sum of even numbers from 1 to 100.
(Hint: Set the start value of the range function to zero and the step value as two.)

Example Output	Sum of even numbers from 1 to 100 : 2550
Time	5min



Write the entire code and the expected output results in the note.

Q4 Use the step value of the range function to find the sum of odd numbers from 1 to 100.
(Hint: Set the start value of the range function to one and the step value as two.)

Example Output	Sum of odd numbers from 1 to 100 : 2500
Time	5min



Write the entire code and the expected output results in the note.

| Pair programming



Pair Programming Practice

| Guideline, mechanisms & contingency plan

Preparing pair programming involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns “driving the mouse.” Effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, it is important to make it clear that the active student will not be punished because the pairing did not work well.

| Pairing similar, not necessarily equal, abilities as partners

Pair programming can be effective when students of similar, though not necessarily equal, abilities are paired as partners. Pairing mismatched students often can lead to unbalanced participation. Teachers must emphasize that pair programming is not a “divide-and-conquer” strategy, but rather a true collaborative effort in every endeavor for the entire project. Teachers should avoid pairing very weak students with very strong students.

| Motivate students by offering extra incentives

Offering extra incentives can help motivate students to pair, especially with advanced students. Some teachers have found it helpful to require students to pair for only one or two assignments.



Pair Programming Practice

| Prevent collaboration cheating

The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

| Collaborative learning environment

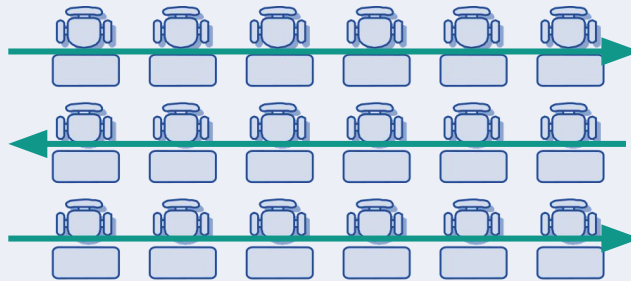
A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

Q1

Agency A is planning to issue tickets of a concert hall for idol singers' concert. Here, the number n is the input and the seat number is arranged as follows. $n * n$ seats are placed when n is given as input. The below arrangement of the seat numbers is called a snake matrix because the array increases in ones in the shape of a snake's trunk. Write a program that produces arrays of these numbers.

Output Example

```
Enter n : 5
 1  2  3  4  5
10  9  8  7  6
11 12 13 14 15
20 19 18 17 16
21 22 23 24 25
```



| Paper coding

Try to fully understand the basic concept before moving on to the next step.

Lack of understanding basic concepts will increase your burden in learning this course, which may make you fail the course.

It may be difficult now, but for successful completion of this course we suggest you to fully understand the concept and move on to the next step.

Q1 Write a program that prints multiplication of 2 using the while statement as following.

**Example
Output**

```
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
```

Time

5min



Write the entire code and the expected output results in the note.

Q2 Let's modify the above program to print all the stages 1 to 9 of the multiplication table. Use only the while statement.

**Example
Output**

```
1*1=1
1*2=2
1*3=3
1*4=4
1*5=5
1*6=6
1*7=7
1*8=8
1*9=9
2*1=2
2*2=4
2*3=6
2*4=8
```

Time

5min



Write the entire code and the expected output results in the note.

| Pair programming



Pair Programming Practice

| Guideline, mechanisms & contingency plan

Preparing pair programming involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns “driving the mouse.” Effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, it is important to make it clear that the active student will not be punished because the pairing did not work well.

| Pairing similar, not necessarily equal, abilities as partners

Pair programming can be effective when students of similar, though not necessarily equal, abilities are paired as partners. Pairing mismatched students often can lead to unbalanced participation. Teachers must emphasize that pair programming is not a “divide-and-conquer” strategy, but rather a true collaborative effort in every endeavor for the entire project. Teachers should avoid pairing very weak students with very strong students.

| Motivate students by offering extra incentives

Offering extra incentives can help motivate students to pair, especially with advanced students. Some teachers have found it helpful to require students to pair for only one or two assignments.



Pair Programming Practice

| Prevent collaboration cheating

The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

| Collaborative learning environment

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

Q1

A palindrome number refers to an integer whose value is the same as its original value, even if listed upside down, such as 121 or 3443. Write the following program to determine whether the number is a palindrome number or not by receiving the number n from the user.

Output Example

```
Enter an integer: 135  
135 is not a palindrome number
```

```
Enter an integer: 3443  
135 is a palindrome number
```

Q2

The computer has a random integer between 1 and 100 as the correct answer value as following. When the user presents the correct answer, the program only informs whether the presented integer is higher or lower compared to the correct answer he or she stored. This game is repeated until the user answers correctly.

Output Example

```
Guess a number between 1 to 100
Enter a number: 50
Lower!
Enter a number: 40
Higher!
Enter a number: 51
Higher!
Enter a number: 45
Lower!
Enter a number: 4
Congratulations. Total try = 5
```