PROCESO GESTIÓN DEL TALENTO HUMANO

FORMATO INFORME MENSUAL EJECUCIÓN CONTRACTUAL

Barranquilla, Mayo de 2023

Señor (a)
JORGE LUIS PEDROZA CABALLERO
SUPERVISOR(A) CONTRATO No. **4521528**
Coordinador Académico Bilinguismo
Bilingüismo
Barranquilla

Asunto: Informe mensual de ejecución contractual
Mes Mayo del año 2023

Referencia: No 4521528 del año 2023

NORLY DE JESUS CALVO ARCHBOLD, identificado con la cédula de ciudadanía No. 32.863.056 de Soledad, en mi calidad de Contratista del SENA, en Bilingüismo, en cumplimiento del Contrato de Prestación de Servicios de la referencia, a continuación presento el Informe de actividades realizadas en el mes objeto de cobro.

Valor y forma de Pago: d). un pago correspondiente al mes de Mayo de 2023 por valor de (3.768.280). Los honorarios serán pagados por el SENA al contratista de acuerdo al cronograma definido por la Dirección Administrativa y Financiera de la Dirección General, en la cuenta de Ahorros No. 474-56853672 Bancolombia, cuyo titular es el Contratista.

Plazo: Será hasta el 30 de Junio de 2023.

OBJETO:
Desarrollar formación profesional por competencias laborales, mediante la estrategia de aprendizaje por proyectos en los programas de formación de FORMACION PRESENCIAL Y VIRTUAL: BILINGUISMO, participando en los equipos de diseño y desarrollo curricular y gestionando las actividades y proyectos a través de las tecnologías de la información y la comunicación que brinda la entidad, atendiendo las políticas institucionales y la normatividad vigente.

Obligaciones Especificas: *(Trascriba las obligaciones especificas del contrato, dentro del siguiente cuadro)*

No	Obligaciones	Acciones realizadas	Evidencias
1	Planificar el desarrollo curricular del programa de formación a impartir de acuerdo a los procedimient os de formación profesional de la entidad.	Revisión de guías de aprendizaje, alistamiento de cursos y elaboración de cronogramas	ADSI FICHAS 2451831, 2404467, 2452524 ADSO FICHAS 2502133, 2638056, 2620654, 2620652, 2558427, 2698057 MTTO EQUIPOS DE COMPUTO FICHAS

			2671178, 2620655 SISTEMAS FICHAS 2671408, 2620629, 2620643 REDES FICHA 2452514
2	Gestionar la información de planificación curricular, gestión de proyectos formativos, juicios evaluativos, novedades e inasistencias en el aplicativo institucional SOFIA Plus	Evaluación de juicios y registro de novedades de cancelamient o de aprendices.	Supervisión coordinación.
3	Gestionar actividades de formación con apoyo en la plataforma virtual dispuesta por la entidad y garantizar el uso de la misma por parte de los aprendices	Procedimient os de alistamiento, ejecución y cierre de cursos de complementa ria virtual con base a los estándares de la formación con modalidad virtual. Evaluar y retroalimentar actividades de aprendizaje por cada unidad temática.	Informe Historial y Centro de Calificaciones. Informes de actividades usuarios en foro y general del curso
4	Ejecutar acciones de formación profesional según el área de conocimiento asignada en el objeto contractual y acorde al nivel de formación, jornada y modalidad establecida en la programació n específica de acciones de formación.	Se atendió a los grupos de aprendices asignados (Bilingüismo)	ADSI FICHAS 2451831, 2404467, 2452524 ADSO FICHAS 2502133, 2638056, 2620654, 2620652, 2558427, 2698057 MTTO EQUIPOS DE COMPUTO FICHAS 2671178, 2620655 SISTEMAS FICHAS 2671408, 2620629, 2620643 REDES FICHA 2452514

A continuación relaciono los desplazamientos que realicé previo a la presentación de este informe. Una vez finalizado cada desplazamiento presenté al ordenador del gasto el informe en el Formato Informe Legalización Desplazamiento Contratista GTH-F-087, en el que se describieron las actividades desarrolladas y los resultados de cada desplazamiento. Cada informe cuenta con el visto bueno del Supervisor.

Se lista a continuación el soporte de la legalización de los desplazamientos realizados, los cuales forman parte integral del presente informe de ejecución contractual.

ITEM	No DE LA ORDEN DE VIAJE	LUGAR DE DESPLAZAMIENTO	FECHAS DE DESPLAZAMIENTOS	
			INICIO	FINALIZACIÓN

Nota 1: *Por cada desplazamiento que haya realizado el contratista, adjuntará el respectivo informe que la soporte. En caso de haber realizado el desplazamiento en fecha posterior a la presentación del informe de ejecución contractual, deberá reportarlo en el siguiente informe de ejecución contractual.*

Para el trámite de la cuenta me permito adjuntar: Documentos electrónicos enunciados como evidencias del cumplimiento de las obligaciones contractuales y los desplazamientos realizados y el No. 9450071422 de la planilla, operador y periodo. (Decreto Ley 2106 de 2019 – " *Decreto Ley Antitrámites"*)

Declaración bajo la gravedad de juramento.

Me permito declarar **bajo la gravedad de juramento** que durante el periodo objeto de cobro SI____NO___ he suscrito otro u otros contratos con entidade(s) del Estado el (los) cual(es) se encuentra(n) en ejecución; la anterior declaración de acuerdo con lo establecido en el numeral 5º de la Directiva Presidencial No. 001 de 2022, a fin de identificar potenciales conflictos de interés y hacer la gestión del riesgos para la toma de decisiones dentro del principio de la debida diligencia frente a la ejecución de los contratos de prestación de servicios profesionales o de apoyo a la gestión.

Nota: En el evento de señalar **SI,** el análisis efectuado por el supervisor y ordenador de pago deberá quedar documentado y publicado en el SECOP II en los documentos relacionados en el archivo Gestión Contractual GC_ del periodo objeto de cobro.

Evidencias en () folios

Cordialmente,

NORLY DE JESUS CALVO ARCHBOLD
Contratista
C.C. No. 32.863.056

Recibí a satisfacción:

JORGE LUIS PEDROZA CABALLERO
Supervisor(a) Contrato 4521528 de 2023
Coordinador Académico Bilinguismo

Plan de clase de programación de software - Nivel básico

Tema: Introducción al vocabulario de programación de software

Objetivos:

1. Familiarizar a los estudiantes con los términos de programación de software en inglés
2. Practicar el uso de los verbos en el presente simple en contextos relacionados con la programación de software
3. Introducir el vocabulario básico de la programación de software

Actividades:

1. Presentación del tema (15 minutos):
   Comenzar la clase presentando el tema a través de preguntas generales sobre la programación de software en inglés. Discutir y anotar las respuestas en la pizarra.

2. Presente simple en contextos de programación de software (20 minutos):
   Introducir y practicar el uso del presente simple en oraciones relacionadas con la programación de software. Preguntar y responder preguntas usando el presente simple, por ejemplo, "What do you do in your programming job?" o "I write code for web applications".

3. Vocabulario básico de programación de software (30 minutos):
   Introducir y discutir los términos básicos de la programación de software en inglés, como "programming language", "variable", "function", "loop", "conditional", "debugging", etc. Repetir los términos y hacer preguntas de comprensión.

4. Actividad de práctica del vocabulario (30 minutos):
   Dar a los estudiantes hojas de trabajo con ejercicios de vocabulario y código de programación para practicar. Los estudiantes deberán identificar las palabras clave en el código y completar los ejercicios.

5. Cierre (5 minutos):
   Repasar los términos y conceptos aprendidos en la clase. Dar a los estudiantes la oportunidad de hacer preguntas y resolver dudas.

Topic: Introduction to programming: basic concepts, data types, variables, and functions.

Vocabulary:

Programming: the process of writing code to create software applications or programs.
Code: the language used to communicate with a computer.
Compiler: a program that converts code into a language that a computer can understand.
Data type: a classification of data that determines how it can be used and what operations can be performed on it.
Variable: a named container for a value that can be changed during the program's execution.
Function: a named block of code that performs a specific task.

More vocabulary

Algorithm - A set of instructions for a computer to perform a task.
Syntax - The rules governing the structure of code in a programming language.
Debugging - The process of finding and fixing errors in code.
Variable - A value that can be assigned and changed in code.
Loop - A block of code that repeats until a specific condition is met.
Function - A block of code that can be called and executed repeatedly.
Comment - A note in code for other programmers to read, but not executed by the computer.

Aquí hay una lista de palabras relacionadas con la syntax y una breve explicación de su uso, junto con algunos ejemplos:

Statement - Una instrucción que realiza una acción en un programa.
Ejemplo: print("Hello World")

Variable - Un espacio en la memoria del ordenador que se utiliza para almacenar valores.
Ejemplo: x = 5

Data type - El tipo de dato que se está utilizando en una variable o en una operación.
Ejemplo: int, float, string, bool

String - Un conjunto de caracteres que se utiliza para representar texto.
Ejemplo: "Hello World"

Integer - Un tipo de datos que representa un número entero.
Ejemplo: 5

F¿loat - Un tipo de datos que representa un número decimal.
Ejemplo: 3.14

Boolean - Un tipo de datos que representa un valor verdadero o falso.
Ejemplo: True, False

Operator - Un símbolo que se utiliza para realizar una operación en una o varias variables.
Ejemplo: +, -, *, /, %, ==, !=, >, <

Expression - Una combinación de variables, operadores y valores que se evalúan para obtener un resultado.
Ejemplo: x + 2

Aquí hay una lista de palabras relacionadas con la syntax en la programación, junto con su explicación y algunas actividades para practicar su uso:

Statement - Una instrucción o acción en un programa. Ejemplo: "Print 'Hello world'" es una declaración.
Actividad:
Escribir cinco declaraciones diferentes para imprimir una serie de números.

Keyword - Una palabra reservada en un lenguaje de programación que tiene un significado especial. Ejemplo: "If" es una palabra clave que se usa para declarar una condición.
Actividad:
Escribir cinco oraciones utilizando las siguientes palabras clave: if, while, for, else, break.

Operator - Un símbolo utilizado para realizar una operación matemática o lógica. Ejemplo: "+" es un operador utilizado para sumar dos números.
Actividad:
Escribir cinco expresiones matemáticas utilizando los siguientes operadores: +, -, *, /.

Function - Un bloque de código que realiza una tarea específica. Ejemplo: "Len()" es una función utilizada para contar el número de caracteres en una cadena.
Actividad:
Escribir cinco funciones diferentes utilizando las siguientes palabras clave: print, input, range, len, max.

Variable - Un contenedor para almacenar datos en un programa. Ejemplo: "Nombre" podría ser una variable utilizada para almacenar el nombre de una persona.
Actividad:
Escribir cinco oraciones utilizando las siguientes variables: nombre, edad, altura, peso, género.

Aquí hay una lista de palabras relacionadas con la syntax y una breve explicación de su uso, junto con algunos ejemplos:

Statement - Una instrucción que realiza una acción en un programa.
Ejemplo: print("Hello World")

Variable - Un espacio en la memoria del ordenador que se utiliza para almacenar valores.
Ejemplo: x = 5

Data type - El tipo de dato que se está utilizando en una variable o en una operación.
Ejemplo: int, float, string, bool

String - Un conjunto de caracteres que se utiliza para representar texto.
Ejemplo: "Hello World"

Integer - Un tipo de datos que representa un número entero.
Ejemplo: 5

Float - Un tipo de datos que representa un número decimal.
Ejemplo: 3.14

Boolean - Un tipo de datos que representa un valor verdadero o falso.
Ejemplo: True, False

Operator - Un símbolo que se utiliza para realizar una operación en una o varias variables.
Ejemplo: +, -, *, /, %, ==, !=, >, <

Expression - Una combinación de variables, operadores y valores que se evalúan para obtener un resultado.
Ejemplo: x + 2

Vocabulario:

Syntax: conjunto de reglas que definen la estructura y el formato correcto de un lenguaje de programación.
Línea de código: una instrucción o comando en el lenguaje de programación.
Palabras clave: palabras reservadas en un lenguaje de programación que tienen un significado específico y no pueden ser utilizadas para otra cosa.
Variables: elementos utilizados para almacenar datos en un programa.
Funciones: un bloque de código que realiza una tarea específica y puede ser llamado varias veces en el programa.
Comentarios: una línea de texto que se utiliza para explicar el código y no se ejecuta en el programa.

Topic: Data Types

Objectives:

Understand the concept of data types in programming
Learn common data types used in programming and their characteristics
Practice using data types in programming exercises
Vocabulary:

Data type
Integer
Float
String
Boolean
Introduction:
Start by explaining to the students what data types are and why they are important in programming. Provide examples of how data types are used in everyday life (e.g. dates, phone numbers, etc.) to make the concept more relatable.

Body:

Explain the four common data types used in programming: integer, float, string, and boolean.
Provide definitions for each data type and their respective characteristics:
Integer: whole numbers without decimals (e.g. 1, 2, 3)
Float: numbers with decimals (e.g. 1.5, 2.75, 3.14)
String: sequence of characters (e.g. "Hello, world!", "1234")
Boolean: true or false (e.g. true, false)
Demonstrate how to declare variables and assign values to each data type using a programming language such as Python.

Provide practice exercises where students have to identify the correct data type to use in a given scenario, or write a program that utilizes different data types.
Worksheet:

Which data type would you use for the following scenarios?
a. Age of a person
b. Temperature of a room
c. Name of a person
d. Is it raining outside?
Write a program that asks for a user's name and age, and prints out a message saying "Hello, [name]! You are [age] years old."

Answers:

a. Integer (intiyer) entero
b. Float (flout) flotante
c. String cadenas de texto
d. Boolean/Bool (bulian) boleanos
Here is an example program in Python:

lua Copy code

```lua
name = input("What is your name? ")
age = input("What is your age? ")
print("Hello, " + name + "! You are " + age + " years old.")
```

Integer: se utilizan para representar números enteros. Ejemplos: 1, 2, 3, -5.

makefile Copy code

```makefile
# Ejemplo en Python
x = 3
y = -2
z = x + y
print(z)  # Output: 1
```

Float: se utilizan para representar números decimales. Ejemplos: 3.14, -0.5, 1.0.

makefile Copy code

```makefile
# Ejemplo en Python
x = 3.14
y = 2.5
z = x + y
print(z)  # Output: 5.64
```

Bool: se utilizan para representar valores de verdadero o falso (True o False). Ejemplos: True, False.

makefile Copy code

```makefile
# Ejemplo en Python
x = 5
y = 3
z = x > y
print(z)  # Output: True
```

String: se utilizan para representar texto o caracteres. Ejemplos: "Hola mundo", "Python es divertido".

bash Copy code

```bash
# Ejemplo en Python
mensaje = "Hola mundo"
print(mensaje)  # Output: Hola mundo
```

Sure! Here are some examples of operators used in programming:

Arithmetic operators: Used to perform basic mathematical operations like addition, subtraction, multiplication, and division. Examples:
Addition: “a + b”
Subtraction: “a – b”
Multiplication: “a * b”
Division: “a / b”

Operadores aritméticos (+, -, *, /, %): estos operadores se utilizan para realizar operaciones matemáticas en valores numéricos. Por ejemplo:

css Copy code

```css
int a = 5;
int b = 2;
int c = a + b;
```

En este caso, la variable "c" tendría un valor de 7 (5 + 2).

Comparison operators: Used to compare two values and return a Boolean value (either true or false) based on the comparison. Examples:

Equal to: “a == b”
Not equal to: “a != b”
Greater than: “a > b”
Less than: “a < b”
Greater than or equal to: “a >= b”
Less than or equal to: “a <= b”

Operadores de comparación (==, !=, >, <, >=, <=): estos operadores se utilizan para comparar dos valores y determinar si son iguales, diferentes, mayores o menores que el otro valor. Por ejemplo:

```arduino
int x = 5;
int y = 3;
bool result = x > y;
```

En este caso, la variable "result" tendría el valor de verdadero (true), ya que 5 es mayor que 3.

Logical operators: Used to combine multiple comparison statements and return a Boolean value. Examples:
And: “a && b”
Or: “a | | b”
Not:” !a”

Operadores lógicos (&&, ||, !): estos operadores se utilizan para combinar dos o más expresiones lógicas y determinar si la expresión compuesta es verdadera o falsa. Por ejemplo:

```arduino
int x = 5;
int y = 3;
bool result = (x > y) && (y != 0);
```

En este caso, la variable "result" tendría el valor de verdadero (true), ya que ambas expresiones son verdaderas.

Assignment operators: Used to assign a value to a variable. Examples:
Assign: “a = b”
Add and assign: “a += b”
Subtract and assign: “a -= b”
Multiply and assign: “a *= b”
Divide and assign: “a /= b”

Operador de asignación (=): este operador se utiliza para asignar un valor a una variable. Por ejemplo:

```arduino
int x = 10;
```

En este caso, el valor de 10 se asigna a la variable "x".

Conditional operator (ternary operator): Used to make a decision based on a condition.
Example: “a ? b : c”
Here's an example activity for practicing operators:

Activity:

Write a program that asks the user for two numbers and performs the following operations:
Adds them together
Subtracts the second number from the first number
Multiplies them together
Divides the first number by the second number and displays the result
Write a program that checks if a number is even or odd using comparison and logical operators.
Worksheet:

For the program in Activity 1, if the user inputs 5 and 2, what will be the output of each operation?

For the program in Activity 2, what will be the output if the user inputs 7? What will be the output if the user inputs 10?

Answers:
1.

Addition: 7
Subtraction: 3
Multiplication: 10
Division: 2.5
2
If the user inputs 7, the output will be "7 is odd."
If the user inputs 10, the output will be "10 is even." Si el usuario ingresa 10, la salida será "10 es par".

Variable (veribol)
Value (valiu)
Odd (aod) impar
Even number (numero par)
Odd number (numero impar)

Daily Routine of a Programmer

A typical daily routine for a programmer might include the following activities:

1. **Check** emails for any important updates or communication from team members or clients.
2. **Attend** daily stand-up meeting with team to discuss progress and plan for the day.
   rite and test code for assigned tasks or projects.
3. **Attend** team meetings to discuss project updates or plan for future sprints.
4. **Collaborate** with team members on projects or code reviews.
5. **Document** code changes or updates.
6. **Attend** training sessions or workshops to learn new programming skills or technologies.

Grammar and Vocabulary

Here are some grammar and vocabulary items related to a programmer's daily routine:

1. Verb tenses: present simple (check, attend), present continuous (writing, testing, collaborating, attending), past simple (documented, attended)
2. Nouns: emails, updates, communication, team, stand-up meeting, progress, code, tasks, projects, code reviews, documentation, training sessions, workshops, programming skills, technologies
3. Adjectives: important, daily, assigned, future, new, programming

Sample Activities

1. Match the verb tense with the activity:

- _____ emails for any important updates or communication from team members or clients. (present simple)
- _____ daily stand-up meeting with team to discuss progress and plan for the day. (present simple)
- _____ and test code for assigned tasks or projects. (present continuous)
- _____ team meetings to discuss project updates or plan for future sprints. (present simple)
- _____ with team members on projects or code reviews. (present continuous)
- _____ code changes or updates. (past simple)
- _____ training sessions or workshops to learn new programming skills or technologies. (present continuous)

1. Fill in the blank with the correct vocabulary item:

- The programmer had to _____ the code for any errors before submitting it.
- The team held a _____ meeting every morning to discuss progress and plan for the day.

- The programmer was assigned several _____ to complete by the end of the week.
- The team had to do _____ of each other's code to ensure quality and consistency.
- The programmer had to _____ all the changes made to the code in the documentation.
- The programmer attended several _____ on new programming languages and tools.

Worksheet with Answers

1. Match the verb tense with the activity:

- Check emails for any important updates or communication from team members or clients. (present simple)
- Attend daily stand-up meeting with team to discuss progress and plan for the day. (present simple)
- Write and test code for assigned tasks or projects. (present continuous)
- Attend team meetings to discuss project updates or plan for future sprints. (present simple)
- Collaborate with team members on projects or code reviews. (present continuous)
- Document code changes or updates. (past simple)
- Attend training sessions or workshops to learn new programming skills or technologies. (present continuous)

2. Fill in the blank with the correct vocabulary item:

- The programmer had to test the code for any errors before submitting it.
- The team held a stand-up meeting every morning to discuss progress and plan for the day.
- The programmer was assigned several tasks to complete by the end of the week.
- The team had to do code reviews of each other's code to ensure quality and consistency.
- The programmer

A day in the life of a programmer:

1. First, **in the morning**, the programmer wakes up and **gets ready for work** by taking a shower and having breakfast.
2. Then, **at 8:30am**, the programmer **commutes to the office** by taking the subway or driving.
3. **Upon arriving** at the office, the programmer **logs into** their computer and **checks their email** for any urgent messages.
4. Next, the programmer **attends** the **morning stand-up meeting** with their team, where they **discuss** the progress of their current project and **share** any obstacles they may be facing.
5. After the meeting, the programmer **sets** their **priorities for the day** and **begins** working on their assigned tasks.
6. Throughout the day, the programmer **collaborates** with their team members, **asks for feedback**, and **debugs** any issues that arise.
7. **In the afternoon**, the programmer may attend a **training session** or **workshop** to learn new skills and technologies.
8. As the day comes to a close, the programmer **finishes** their **current task**, **saves their work**, and **logs out** of their computer.
9. **Finally**, the programmer **packs up** their belongings and **heads home** to relax and unwind.

Activity:

1. Read the above daily routine and identify the sequence adverbs and prepositions used.
2. Write a similar daily routine for a programmer working remotely.
3. Fill in the blanks with the appropriate sequence adverb or preposition: "The programmer starts working ___ their tasks ___ the morning."
4. Discuss with a partner the importance of collaborating and asking for feedback in a team environment.
5. Write a paragraph about your own daily routine and the sequence adverbs and prepositions you use.

Worksheet with Answers:

1. Sequence adverbs: first, then, next, throughout the day, in the afternoon, finally. Prepositions: in the morning, at 8:30am, upon arriving, for the day, with their team, in a team environment, throughout the day, as the day comes to a close, to relax and unwind.
2. Example daily routine for a programmer working remotely:
    - At 9am, the programmer logs into their computer and checks their email.
    - Then, they review their assigned tasks and set their priorities for the day.
    - Throughout the day, the programmer communicates with their team members through video conferences and instant messaging.
    - In the afternoon, the programmer takes a break and goes for a walk to stretch their legs.
    - Finally, at 5pm, the programmer logs out of their computer and enjoys their free time.

Daily Routine of a Programmer

Complete the following sentences with the correct sequence expressions, prepositions, and vocabulary.

1. I usually wake _____ at 7:00 AM and immediately check my email for any urgent messages.
2. After checking my email, I take a quick shower and get _____ dressed for work.
3. I leave my house _____ 8:00 AM and walk to the bus stop.
4. I take the bus to work and usually arrive at the office _____ 9:00 AM.
5. Once I arrive, I grab a cup of coffee and check in with my team to see what tasks need to be _____ first.
6. After prioritizing my tasks, I begin working _____ them one by one.
7. I always take a lunch break _____ 12:00 PM and usually bring my own food from home.
8. In the afternoon, I attend any necessary meetings or collaborate with my team _____ any issues that may have come up.
9. As the end of the workday approaches, I start wrapping up any outstanding tasks _____ completing any necessary documentation.
10. I leave the office _____ 5:00 PM and usually walk home to get some exercise.

Answers:

1. up
2. properly
3. at
4. around
5. tackled

6. on
7. at
8. on
9. by
10. at

Sure, here's an example of a daily routine of a programmer using present simple, sequence adverbs, prepositions, and relevant vocabulary:

Daily Routine of a Programmer

John is a programmer. He **wakes up** at 6:00 am every day and **gets out of bed**. He **goes to** the kitchen and **makes** himself **a cup of coffee**. Then, he **sits down** at his computer and **checks** his emails.

At 8:00 am, John **starts working** on his project. He **writes code**, **tests** it, and **debugs** any errors he finds. He **takes a break** at 12:00 pm for lunch. He usually **eats** a sandwich or salad at his desk.

After lunch, John **continues working** until 6:00 pm. He **saves** his work, **shuts down** his computer, and l**eaves** his home office. He **goes to the gym** to exercise for an hour. Then, he **comes back** home and **has dinner**. He **spends** the rest of the evening **relaxing** and **watching TV**.

Worksheet

What time does John wake up every day?
a. 7:00 am
b. 8:00 am
c. 6:00 am

What does John do after he wakes up?
a. He goes to the gym
b. He checks his emails
c. He eats breakfast

What does John do at 12:00 pm?
a. He takes a break for lunch
b. He goes for a walk
c. He calls his friends

What does John do after work?
a. He goes to the gym
b. He watches TV
c. He goes out with friends

What does John do in the evening?
a. He works on his project
b**.** He watches TV
c. He goes to bed

Topic: Writing instructions in programming
Vocabulary:

- Syntax
- Code
- Algorithm
- Debugging
- Compiler
- Command
- Loop
- Function
- Parameter

Grammar:

- Imperative sentences
- Simple present tense
- Modal verbs (can, must)
- Adverbs of frequency (always, usually, often, sometimes, rarely, never)
- Prepositions (in, at, on, with)

Connectors of sequence:

- First, second, third
- Next
- Then
- After that
- Finally
- In the end

Example activity:

1. Write a simple algorithm for adding two numbers in Python. Use imperative sentences and the simple present tense. (Example: "Enter the first number. Add it to the second number. Print the result.")
2. Use the algorithm to write a Python program. Use the vocabulary and connectors of sequence to write clear and concise instructions for the user.
3. Have students exchange programs with a partner and test them out. Encourage them to use debugging techniques to find and fix any errors.

Example worksheet:
Instructions for Adding Two Numbers in Python

1. First, open a new Python file.
2. Next, write the following code to define a function that will add two numbers:

```python
def add_numbers(num1, num2):
    sum = num1 + num2
    return sum
```

3. After that, write a command to prompt the user to enter the first number.
4. Then, write a command to store the first number in a variable.
5. Next, write a command to prompt the user to enter the second number.
6. Then, write a command to store the second number in a variable.
7. After that, write a command to call the **add_numbers** function with the two numbers as parameters.
8. Then, write a command to print the result of the addition.
9. Finally, in the end, save the file and run it through a compiler.

Response:

1. Enter the first number: 3
2. Enter the second number: 4
3. The sum of 3 and 4 is 7.

Topic: Giving instructions in programming using imperatives, present simple, connectors of sequence, prepositions, and relevant vocabulary.

Vocabulary:

- Algorithm
- Code
- Debugging
- Function
- Loop
- Syntax
- Variable

Grammar:

- Imperatives
- Present Simple
- Connectors of sequence (First, Then, Next, Finally)
- Prepositions (In, On, At)

Example Activities:

1. Instructions Writing Exercise: Ask students to write step-by-step instructions on how to make a sandwich using the following vocabulary words: bread, cheese, lettuce, tomato, and mayonnaise. Afterward, ask them to exchange instructions with a partner, read and follow the instructions to make the sandwich.
2. Programming Instructions Exercise: Provide students with a short program that needs to be debugged. Ask them to work in pairs and write step-by-step instructions on how to debug the program using the relevant vocabulary words and grammar structures. Afterward, ask them to

exchange their instructions with another pair, read and follow the instructions to debug the program.

3. Fill in the Blank Exercise: Provide students with a short paragraph containing gaps that they need to fill in with the appropriate preposition or connector of sequence.

Example Paragraph: ___ programming, it's important to use the correct ___ in order to create an algorithm that works. ___ writing a function, it's essential to use the right ___ and pay attention to the ___ of the code. ___ the end of the program, it's important to run a debugging tool ___ order to catch any errors.

Worksheet:

Instructions Writing Exercise: Write step-by-step instructions on how to create a loop using the "for" statement in Python. Use the following vocabulary words: algorithm, syntax, variable, code, and loop.

1. ___ a new file in your text editor of choice.
2. ___ the file a name and save it ___ your computer.
3. ___ by writing the "for" statement, followed by a ___ in parentheses.
4. ___ the variable that you want to use in the loop.
5. ___ the algorithm that you want to use in the loop ___ the code block.
6. ___ the loop by adding the necessary syntax to ___ the loop.
7. ___ the code ___ errors and fix them ___ necessary.

Answers:

1. Open
2. Give, on
3. Start, semicolon
4. Define
5. Write, inside
6. End, close
7. Check, and, if

Activity 1: Complete the instructions

Instructions:

1. _____ the program.
2. _____ the code.
3. _____ the output.
4. _____ the bugs.
5. _____ the program again.

Words to choose from: run, fix, test, write, check

Worksheet 1 - Responses:

1. Run the program.
2. Write the code.
3. Test the output.
4. Fix the bugs.
5. Check the program again.

Activity 2: Write your own instructions

Instructions:

1. Write a program that does the following:

2. _____ the user for input.
3. _____ the input into the program.
4. _____ the calculations.
5. _____ the output.
6. _____ the program.

Worksheet 2 - Responses:

1. Write a program that does the following:
2. Ask the user for input.
3. Import the input into the program.
4. Perform the calculations.
5. Output the results.
6. End the program.

Activity 3: Correct the instructions

Instructions:

1. Open the program.
2. Click the "run" button.
3. Fix the errors.
4. Save the program.
5. Repeat step 2.

Worksheet 3 - Responses:

1. Open the program.
2. Click the "run" button.
3. Fix the errors.
4. Save the program.
5. Repeat step 2.

Activity 4: Put the instructions in order

Instructions:

1. _____ the program.
2. _____ the code.
3. _____ the output.
4. _____ the bugs.
5. _____ the program again.

Words to choose from: run, fix, test, write, check

Worksheet 4 - Responses:

1. Write the code.
2. Test the output.
3. Fix the bugs.
4. Check the program again.
5. Run the program.

Activity 5: Match the instructions

Instructions:

1. _____ the program.
2. _____ the bugs.
3. _____ the code.

4. _____ the output.
5. _____ the program again.

Words to choose from: run, fix, test, write, check

Worksheet 5 - Responses:

1. Run the program.
2. Fix the bugs.
3. Write the code.
4. Test the output.
5. Check the program again.

Sure! Here are some examples of activities and worksheets on the topic of giving instructions in programming using imperatives, present simple, and relevant vocabulary:

Activity 1: Vocabulary matching

Instructions:

Match the programming vocabulary on the left with its definition on the right.

1. Loop a. A named section of code that performs a specific task
2. Function b. A set of instructions that is repeated until a certain condition is met
3. Variable c. A sequence of instructions that is executed when a specific event occurs
4. Conditional d. A named storage location that holds a value that can be changed
5. Event e. A statement that performs an action if a certain condition is met

Worksheet:

1. A set of instructions that is repeated until a certain condition is met
2. A named section of code that performs a specific task
3. A named storage location that holds a value that can be changed
4. A sequence of instructions that is executed when a specific event occurs
5. A statement that performs an action if a certain condition is met

Answer Key:

1. b
2. a
3. d
4. c
5. e

Activity 2: Writing instructions

Instructions:

Write instructions for a program that takes in a user's name and age and prints out a message that says "Hello, [Name]. You are [Age] years old."

Worksheet:

1. Use the input() function to ask the user for their name and store it in a variable.
2. Use the input() function to ask the user for their age and store it in a variable.
3. Use the print() function to output the message "Hello, [Name]. You are [Age] years old.", where [Name] and [Age] are replaced with the values stored in the variables.

Answer Key:

1. name = input("What is your name? ")
2. age = input("How old are you? ")

3. print("Hello, " + name + ". You are " + age + " years old.")

Activity 3: Fill in the blanks

Instructions:

Fill in the blanks with the correct form of the verb in parentheses.

1. First, _____ (open) the Python editor.
2. Next, _____ (type) the program code.
3. Then, _____ (run) the program to see the output.
4. Finally, _____ (save) the program to your computer.

Worksheet:

1. open
2. type
3. run
4. save

Answer Key:

1. open
2. type
3. run
4. save

Bucle / Loop
Condición / Condition
Variable / Variable
Función / Function
Clase / Class

Ejemplo de términos de programación para esta actividad:

Términos en español	**Traducción en inglés**
Bucle	Loop
Condición	Condition
Variable	Variable
Función	Function
Clase	Class

Ejemplo de términos de programación para esta actividad:

Definición	**Término en inglés**
A piece of code that can be reused	Function
A conditional statement that performs different actions depending on whether a certain condition is met	If Statement
A sequence of instructions that is continually repeated	Loop
A storage location that holds a value	Variable
A template for creating objects	Class

Ejemplo de oraciones para esta actividad:

I used a loop to repeat the code.
The condition statement checks if something is true or false.
A variable is a container that holds a value.
A function is a block of code that can be reused.
A class is a blueprint for creating objects.

Ejemplo de términos de programación y oraciones para esta actividad:

Términos de programación	
If Statement	Function
Loop	Variable
Class	Conditional Statement

A _____ is a block of code that can be reused in a program.
A _____ is a storage location that holds a value.
A _____ is a template for creating objects in object-oriented programming.
An _____ is a conditional statement that performs different actions depending on whether a certain condition is met.
A _____ is a sequence of instructions that is continually repeated until a certain condition is met.

Respuestas:

1. A **Function** is a block of code that can be reused in a program.
2. A **Variable** is a storage location that holds a value.
3. A **Class** is a template for creating objects in object-oriented programming.
4. An **If Statement** is a conditional statement that performs different actions depending on whether a certain condition is met.
5. A **Loop** is a sequence of instructions that is continually repeated until a certain condition is met.

Ejemplo de términos de programación y definiciones para esta actividad:

Términos de programación	**Definiciones**
Variable	A statement that defines a group of objects
Function	A condition that tests for equality
If Statement	A storage location that holds a value
Loop	A block of code that performs a specific task
Class	A block of code that can be reused in a program
Method	A named sequence of statements
Object	A template for creating objects in object-oriented programming
Array	A sequence of instructions that is continually repeated until a certain condition is met
Operator	A set of data values that cannot be changed
Boolean	A data type that represents true or false
String	A data type that represents a sequence of characters
Integer	A data type that represents a whole number
Float	A data type that represents a decimal number
Double	A data type that represents a double-precision decimal number
Char	A data type that represents a single character
Void	A data type that represents the absence of a value
Exception	A block of code that can be executed if an error occurs
Debugging	A process of finding and fixing errors in a program

Términos de programación	Definiciones
IDE	A software application for coding and debugging
Compiler	A program that translates code into machine-readable format

Respuestas:

Términos de programación	Definiciones
Variable	A storage location that holds a value
Function	A named sequence of statements
If Statement	A condition that tests for equality
Loop	A sequence of instructions that is continually repeated until a certain condition is met
Class	A template for creating objects in object-oriented programming
Method	A block of code that performs a specific task
Object	A group of related variables and methods
Array	A statement that defines a group of objects
Operator	A symbol used to perform a mathematical or logical operation
Boolean	A data type that represents true or false
String	A data type that represents a sequence of characters
Integer	A data type that represents a whole number
Float	A data type that represents a decimal number
Double	A data type that represents a double-precision decimal number
Char	A data type that represents a single character
Void	A data type that represents the absence of a value
Exception	A block of code that can be executed if an error occurs
Debug	

Ejemplo de términos de programación y significado en español para esta actividad:

Términos de programación	Significado en español
Variable	Variable
Function	Función
If Statement	Sentencia If
Loop	Bucle
Class	Clase
Method	Método

Términos de programación	Significado en español
Object	Objeto
Array	Matriz
Operator	Operador
Boolean	Booleano
String	Cadena de caracteres
Integer	Número entero
Float	Número decimal
Double	Número decimal de doble precisión
Char	Carácter

Respuestas:

Términos de programación	Significado en español
Variable	Variable
Function	Función
If Statement	Sentencia If
Loop	Bucle
Class	Clase
Method	Método
Object	Objeto
Array	Matriz
Operator	Operador
Boolean	Booleano
String	Cadena de caracteres
Integer	Número entero
Float	Número decimal
Double	Número decimal de doble precisión
Char	Carácter

Términos de programación	Definiciones en ingles	Definiciones en español
Variable	A storage location that holds a value	Variable
Function	A named sequence of statements	Function
If Statement	A condition that tests for equality	Sentencia if
Loop	A sequence of instructions that is continually repeated until a certain condition is met	Bucle
Class	A template for creating objects in object-oriented programming	Clase
Method	A block of code that performs a specific task	Metodo
Object	A group of related variables and methods	Objeto
Array	A statement that defines a group of objects	Matriz
Operator	A symbol used to perform a mathematical or logical operation	Operador
Boolean	A data type that represents true or false	Booleano
String	A data type that represents a sequence of characters	Cadena de caracteres
Integer	A data type that represents a whole number	Numero entero
Float	A data type that represents a decimal number	Numero decimal
Query	A request for information from a database using a specific language, such as SQL (Structured Query Language).	Una pregunta que se realiza a una base de datos para obtener información.
Char	A data type that represents a single character	caracter
Programming language	A formal language used to communicate with computers and create programs, applications, and software	Lenguaje de programacion
Database	A collection of data that is organized in a way that can be easily accessed, managed, and updated.	Base de datos
Debugging	A process of finding and fixing errors, bugs, or defects in software code.	Depuracion
GUI	stands for "Graphical User Interface," and it refers to the visual interface of a program that allows users to interact with it using visual elements such as icons, buttons, and menus.	Interfaz Grafica de Usuario
Syntax	A set of rules that govern the structure and organization of programming languages, including the arrangement of keywords, symbols, and punctuation marks. Proper syntax is necessary for a program to be executed correctly.	Sintaxis

Worksheet "Programación en Inglés":

Algorithm
a) Un conjunto de reglas para resolver un problema en un número finito de pasos.
b) Una estructura de datos que contiene una colección de elementos, cada uno identificado por un índice.
c) Instrucciones escritas en un lenguaje de programación para que una computadora las ejecute.

Array
a) Un error o problema en un programa que causa un comportamiento inesperado.
b) Un tipo de dato que puede ser verdadero o falso.
c) Una estructura de datos que contiene una colección de elementos, cada uno identificado por un índice.

Boolean
a) Un tipo de dato que puede ser verdadero o falso.
b) Un error o problema en un programa que causa un comportamiento inesperado.
c) Un programa que traduce el código fuente de un programa a lenguaje de máquina para que la computadora lo pueda ejecutar.

Bug
a) El proceso de encontrar y corregir errores en un programa.
b) Instrucciones escritas en un lenguaje de programación para que una computadora las ejecute.
c) **Un error o problema en un programa que causa un comportamiento inesperado.**

Class
a) Una plantilla o molde para crear objetos que comparten los mismos atributos y métodos.
b) Instrucciones escritas en un lenguaje de programación para que una computadora las ejecute.
c) El proceso de encontrar y corregir errores en un programa.

Code
a) Instrucciones escritas en un lenguaje de programación para que una computadora las ejecute.
b) Un programa que traduce el código fuente de un programa a lenguaje de máquina para que la computadora lo pueda ejecutar.
c) Un bloque de código que realiza una tarea específica y puede ser llamado desde otras partes del programa.

Compiler
a) Instrucciones escritas en un lenguaje de programación para que una computadora las ejecute.
b) El proceso de encontrar y corregir errores en un programa.
c) Un programa que traduce el código fuente de un programa a lenguaje de máquina para que la computadora lo pueda ejecutar.

Debugging
a) Una estructura que define un conjunto de métodos y propiedades que un objeto debe implementar.
b) Instrucciones escritas en un lenguaje de programación para que una computadora las ejecute.
c) El proceso de encontrar y corregir errores en un programa

Function

a) Un bloque de código que realiza una tarea específica y puede ser llamado desde otras partes del programa.
b) Un conjunto de instrucciones escritas en un lenguaje de programación que realizan una tarea específica.
c) Una instancia de una clase que tiene atributos y métodos.

Interface

a) Un símbolo o palabra reservada que realiza una operación en uno o más valores.
b) Una estructura que define un conjunto de métodos y propiedades que un objeto debe implementar.
c) Un bloque de código que se ejecuta repetidamente mientras se cumpla una condición.

Loop

a) Una instancia de una clase que tiene atributos y métodos.
b) Un símbolo o palabra reservada que realiza una operación en uno o más valores.
c) **Un bloque de código que se ejecuta repetidamente mientras se cumpla una condición.**

Object

a) Una instancia de una clase que tiene atributos y métodos.
b) Un archivo que contiene instrucciones que pueden ser ejecutadas por una computadora.
c) Una secuencia de caracteres, como letras, números y símbolos.

Operator

a) **Un símbolo o palabra reservada que realiza una operación en uno o más valores.**
b) Un conjunto de instrucciones escritas en un lenguaje de programación que realizan una tarea específica.
c) Las reglas que definen la estructura y el formato de un lenguaje de programación.

Program

a) Un conjunto de instrucciones escritas en un lenguaje de programación que realizan una tarea específica.
b) Un contenedor que almacena un valor que puede cambiar durante la ejecución de un programa.
c) Un sistema que permite el seguimiento de cambios en el código fuente de un programa y la colaboración en su desarrollo.

Script

a) Una secuencia de caracteres, como letras, números y símbolos.
b) Un archivo que contiene instrucciones que pueden ser ejecutadas por una computadora.
c) Las reglas que definen la estructura y el formato de un lenguaje de programación.

String

a) Un sistema que permite el seguimiento de cambios en el código fuente de un programa y la colaboración en su desarrollo.
b) Las reglas que definen la estructura y el formato de un lenguaje de programación.
c) **Una secuencia de caracteres, como letras, números y símbolos.**

Syntax

a) Un bucle que se ejecuta mientras se cumple una condición especificada.
b) Un sistema que permite el seguimiento de cambios en el código fuente de un programa y la colaboración en su desarrollo.
c) Las reglas que definen la estructura y el formato de un lenguaje de programación.

Variable

a) Una instancia de una clase que tiene atributos y métodos.
b) Un símbolo o palabra reservada que realiza una operación en uno o más valores.
c) **Un contenedor que almacena un valor que puede cambiar durante la ejecución de un programa.**

Version Control

a) Un programa que traduce el código fuente de un programa a lenguaje de máquina para que la computadora lo pueda ejecutar.
b) Un bloque de código que realiza una tarea específica y puede ser llamado desde otras partes del programa.
c) **Un sistema que permite el seguimiento de cambios en el código fuente de un programa y la colaboración en su desarrollo.**

While Loop

a) **Un bucle que se ejecuta mientras se cumple una condición especificada.**
b) Una estructura de datos que contiene una colección de elementos, cada uno identificado por un índice.
c) Instrucciones escritas en un lenguaje de programación para que una computadora las ejecute.

Términos de programación	**Definiciones en español**
Variable	
Function	
If Statement	
Loop	
Class	
Method	
Object	
Array	
Operator	
Boolean	
String	
Integer	
Float	
Query	
Char	
Programming language	
Database	
Debugging	
GUI	
Syntax	

Instrucciones: Relaciona cada término de programación en inglés con su correspondiente definición en español.

1. Algorithm (f)	a. Programa que convierte código fuente en código objeto.
2. Code (c)	b. Proceso de encontrar y arreglar errores en el código.
3. Compiler (a)	c. El proceso de escribir y mantener código de programación.
4. Debugging (b)	d. Un conjunto de instrucciones que realizan una tarea específica.
5. Function (d)	e. Una herramienta que permite al usuario interactuar con un programa.
6. GUI (e)	f. Un conjunto de pasos que resuelve un problema específico.
7. Interface (t)	g. Un tipo de ciclo que se repite hasta que se cumple una condición.
8. Input (s)	h. Un tipo de datos que solo puede tener dos valores: Verdadero o Falso.
9. Loop (g)	i. Una colección de datos organizados en una estructura específica.
10. Object (j)	j. Un conjunto de elementos de programación que se pueden reutilizar.
11. Output (k)	k. Una parte de un programa que recibe entradas y produce salidas.
12. Parameter (q)	l. Un conjunto de reglas que definen cómo se escribe un programa.
13. Programming language (m)	m. Un conjunto de elementos que se utilizan para escribir código.
14. Query (n)	n. Una pregunta que se realiza a una base de datos para obtener información.
15. Script (o)	o. Una secuencia de comandos que se ejecutan automáticamente.
16. Syntax (r)	p. El proceso de diseñar y construir sitios web y aplicaciones web.
17. Variable (i)	q. Un valor que se proporciona a una función cuando se llama.
18. Web development (p)	r. La estructura de un programa, incluyendo la sintaxis y las convenciones.
19. Boolean (h)	s. Datos que se ingresan en un programa para su procesamiento.
20. Database. (u)	t. Un conjunto de información organizada en un sistema de recuperación.

Worksheet de Vocabulario de Programación en Inglés

1. Algorithm: _____
2. Array: _____
3. Boolean: _____
4. Bug: _____
5. Class: _____
6. Code: _____
7. Compiler: _____
8. Debugging: _____
9. Function: _____
10. Interface: _____
11. Loop: _____
12. Object: _____
13. Operator: _____
14. Parameter: _____
15. Source code: _____
16. Syntax: _____
17. Variable: _____
18. Code block: _____
19. Comment: _____
20. Inheritance: _____

Worksheet de Vocabulario de Programación en Inglés

1. Algorithm: _____
2. Array: _____
3. Boolean: _____
4. Bug: _____
5. Class: _____
6. Code: _____
7. Compiler: _____
8. Debugging: _____
9. Function: _____
10. Interface: _____
11. Loop: _____
12. Object: _____
13. Operator: _____
14. Parameter: _____
15. Source code: _____
16. Syntax: _____
17. Variable: _____
18. Code block: _____
19. Comment: _____
20. Inheritance: _____

Worksheet de Vocabulario de Programación en Inglés

1. Algorithm: _____
2. Array: _____
3. Boolean: _____
4. Bug: _____
5. Class: _____
6. Code: _____
7. Compiler: _____
8. Debugging: _____
9. Function: _____
10. Interface: _____
11. Loop: _____
12. Object: _____
13. Operator: _____
14. Parameter: _____
15. Source code: _____
16. Syntax: _____
17. Variable: _____
18. Code block: _____
19. Comment: _____
20. Inheritance: _____

Worksheet de Vocabulario de Programación en Inglés

1. Algorithm: _____
2. Array: _____
3. Boolean: _____
4. Bug: _____
5. Class: _____
6. Code: _____
7. Compiler: _____
8. Debugging: _____
9. Function: _____
10. Interface: _____
11. Loop: _____
12. Object: _____
13. Operator: _____
14. Parameter: _____
15. Source code: _____
16. Syntax: _____
17. Variable: _____
18. Code block: _____
19. Comment: _____
20. Inheritance: _____

Directions: Fill in the blank with the correct form of the present simple of the verb in parentheses.

Java (be) _____ a popular programming language.
The program (run) _____ without errors.
The software (use) _____ a simple user interface.
The programmer (write) _____ code every day.
The debugger (fix) _____ errors in the code.
The database (store) _____ user information.
The algorithm (solve) _____ complex problems.
The software engineer (test) _____ new features.
The client (request) _____ a custom software solution.
The user (access) _____ the program through a web browser.

Directions: Fill in the blank with the correct form of the present simple of the verb in parentheses.

Java (be) _____ a popular programming language.
The program (run) _____ without errors.
The software (use) _____ a simple user interface.
The programmer (write) _____ code every day.
The debugger (fix) _____ errors in the code.
The database (store) _____ user information.
The algorithm (solve) _____ complex problems.
The software engineer (test) _____ new features.
The client (request) _____ a custom software solution.
The user (access) _____ the program through a web browser.

Directions: Fill in the blank with the correct form of the present simple of the verb in parentheses.

Java (be) _____ a popular programming language.
The program (run) _____ without errors.
The software (use) _____ a simple user interface.
The programmer (write) _____ code every day.
The debugger (fix) _____ errors in the code.
The database (store) _____ user information.
The algorithm (solve) _____ complex problems.
The software engineer (test) _____ new features.
The client (request) _____ a custom software solution.
The user (access) _____ the program through a web browser.

Completar oraciones con el verbo en presente simple 1. Juan, Miguel and I _____ (code) in Java. 2. Jonathan _____ (debug) his program. 3. Sandra and Melany _____ (test) the software. 4. Katherine _____ (program) the new feature. 5. I _____ (develop) mobile applications.	**Completar oraciones con el verbo en presente simple** 1. Juan, Miguel and I _____ (code) in Java. 2. Jonathan _____ (debug) his program. 3. Sandra and Melany _____ (test) the software. 4. Katherine _____ (program) the new feature. 5. I _____ (develop) mobile applications.
Completar oraciones con el verbo en presente simple 1. Juan, Miguel and I _____ (code) in Java. 2. Jonathan _____ (debug) his program. 3. Sandra and Melany _____ (test) the software. 4. Katherine _____ (program) the new feature. 5. I _____ (develop) mobile applications.	**Completar oraciones con el verbo en presente simple** 1. Juan, Miguel and I _____ (code) in Java. 2. Jonathan _____ (debug) his program. 3. Sandra and Melany _____ (test) the software. 4. Katherine _____ (program) the new feature. 5. I _____ (develop) mobile applications.
Completar oraciones con el verbo en presente simple 1. Juan, Miguel and I _____ (code) in Java. 2. Jonathan _____ (debug) his program. 3. Sandra and Melany _____ (test) the software. 4. Katherine _____ (program) the new feature. 5. I _____ (develop) mobile applications.	**Completar oraciones con el verbo en presente simple** 1. Juan, Miguel and I _____ (code) in Java. 2. Jonathan _____ (debug) his program. 3. Sandra and Melany _____ (test) the software. 4. Katherine _____ (program) the new feature. 5. I _____ (develop) mobile applications.

Completar diálogos con el verbo en presente simple

A: What do you do?
B: I ____ (be) a software engineer.
A: And what about you?
B: I _____ (develop) applications.
A: That's interesting. And who _____ (to test) the software?
B: My workmate _____ (to test) it.

Escribir oraciones en presente simple sobre programación de software
Escribe oraciones en presente simple sobre los siguientes temas:

1. Programming languages
2. Debugging
3. Software testing
4. Mobile applications

Respuestas:

1. Python is a popular programming language.
2. He debugs his code to find errors.
3. They test the software before releasing it.
4. I develop mobile applications for iOS and Android.

Completa el párrafo: Presenta un párrafo que contenga varios verbos en el presente simple relacionados con la programación de software, pero algunos verbos están faltantes. Pide a los estudiantes que completen el párrafo usando la forma correcta del verbo. Por ejemplo:

"A programmer _____ (write) code using programming languages such as Java and Python. They _____ (debug) code to find errors and _____ (test) the code to make sure it works properly. Finally, they _____ (upload) the code to a database or version control system."

Respuestas: writes, debugs, tests, uploads

Verbos: Debug, Write, Test, Upload
Oraciones:
The programmer uploads the code to GitHub.
The programmer tests the software to make sure it works properly.
The programmer writes code using Python.
The programmer debugs the code to find errors.
Respuestas:
Upload, Test, Write, Debug

ejemplos de actividades con el uso del presente simple en contexto, junto con sus respuestas:

Actividad 1: Complete the sentences with the correct form of the present simple.

She _____ (work) at a software company.
They _____ (study) computer science.
He _____ (code) in Java.
We _____ (use) a programming language.
You _____ (write) code every day.

Actividad 2: Choose the correct form of the present simple to complete the sentences.

My computer _____ (run / runs) very slowly.
The software _____ (is / are) easy to use.
The program _____ (have / has) a lot of bugs.
She always _____ (forget / forgets) to save her work.
We _____ (enjoy / enjoys) learning about new technologies.

Actividad 3: Rewrite the sentences using the present simple.

I am working on a new project.
I _____ (work) on a new project.
He is using a Macbook Pro for programming.
He _____ (use) a Macbook Pro for programming.
They are studying Python at university.
They _____ (study) Python at university.
She is not coding in C++.
She _____ (not code) in C++.
We are not enjoying the class.
We _____ (not enjoy) the class.

ejemplos de actividades con el uso del presente simple en contexto para estudiantes de informática de nivel intermedio:

Actividad 1: Completar oraciones con el verbo en presente simple
Completa las siguientes oraciones con el verbo en presente simple que corresponda:

The computer _____ (to crash) frequently.
Our team _____ (to meet) every Monday to discuss the project progress.
The software _____ (to update) automatically every week.
She always _____ (to write) clean and efficient code.
The IT department _____ (to handle) all the technical issues of the company.

Actividad 2: Verdadero o falso
Lee las siguientes oraciones y decide si son verdaderas o falsas:

The internet is slow today. (Falso)
Our software supports multiple languages. (Verdadero)
They doesn't know how to code in Python. (Falso)
He usually arrives to work late. (Falso)
The server crashes every time we run a certain query. (Verdadero)

Actividad 3: Selección múltiple
Elige la opción correcta para completar las siguientes oraciones:

The development team _____ agile methodology.
a) use
b) uses
c) using

The computer system _____ down for maintenance every Friday at 8 pm.
a) shut
b) shuts
c) shutting

Our company _____ in the field of artificial intelligence.
a) specialize
b) specializes
c) specializing

The software _____ all the data entered by the user.
a) store
b) stores
c) storing

The IT support team _____ tickets via email or phone.
a) handle
b) handles
c) handling

ORACIONES IMPERATIVO

Eat a healthy breakfast every day.
Study for your exams every day.
Write clean and readable code.
Use comments in your code.
Test your code thoroughly before submitting.
Take regular breaks to avoid burnout.
Don't forget to double-check your work.

Write clear and concise comments in your code.
Check your code for errors before submitting it.
Pay attention to detail when programming.
Practice programming every day to improve your skills.
Read the documentation before using a new tool or library.

Actividad 2:
Lea la siguiente lista de tareas para un proyecto de programación. Escriba oraciones en imperativo para cada tarea.

Choose a programming language.
Design the user interface.
Write the code.
Test the program.
Fix any bugs.
Deploy the program.

Actividad 1: Ordenando instrucciones
Reordena las palabras para formar instrucciones en imperativo y completa las instrucciones con los verbos adecuados. Luego, escribe las instrucciones completas en la forma correcta.

Ejemplo: this / file / save / please. - Save this file, please.

to the library / go / now / let's - _____
your email / check / regularly / please - _____
the program / install / please - _____
the printer / turn on / please - _____
your code / test / now / let's - _____
Respuestas:

Let's go to the library now.
Please check your email regularly.
Please install the program.
Please turn on the printer.
Let's test your code now.

Actividad 2: Instrucciones para solucionar problemas
Completa las siguientes instrucciones en imperativo con los verbos adecuados para solucionar los problemas dados.

Ejemplo: I can't log in. - Check your username and password.

I can't find my document. - _____
My computer keeps freezing. - _____
I can't connect to the internet. - _____
My computer is running slow. - _____
I accidentally deleted my file. - _____
Respuestas:

Check your recent documents.
Restart your computer.
Check your internet connection.
Close unnecessary programs and files.
Check your recycle bin or backup files.

Actividad 3: Dar consejos
Completa las siguientes oraciones con los verbos adecuados en imperativo para dar consejos sobre informática.

Ejemplo: You should back up your important files. - Back up your important files.

You need to update your software regularly. - _____
You should create strong passwords. - _____
You need to scan your computer for viruses regularly. - _____
You should organize your files and folders. - _____
You need to be careful with suspicious emails. - _____
Respuestas:

Update your software regularly.
Create strong passwords.
Scan your computer for viruses regularly.
Organize your files and folders.
Be careful with suspicious emails.

ejemplos de actividades con el uso del imperativo en inglés en contexto para estudiantes de redes:

Actividad 1:
Completa las siguientes instrucciones con el verbo en imperativo correspondiente:

Ejemplo: _____ (Open) the command prompt. - Open

_____ (Connect) the cable to the router.
_____ (Type) your username and password.
_____ (Restart) the computer.
_____ (Ping) the IP address.
_____ (Configure) the network settings.
Respuestas:

Connect
Type
Restart
Ping
Configure
Actividad 2:
Escribe una lista de instrucciones para configurar una red inalámbrica en un dispositivo móvil utilizando el imperativo. Incluye al menos 5 pasos y utiliza los siguientes verbos: turn on, select, enter, choose, connect.

Ejemplo de respuesta:

Turn on your device's Wi-Fi.
Select the Wi-Fi network you want to connect to.
Enter the password for the Wi-Fi network.
Choose to connect automatically to this network.
Connect to the Wi-Fi network.

Actividad 3:
Lee el siguiente diálogo y completa las oraciones con el verbo en imperativo correspondiente:

A: "I'm having trouble accessing the internet on my computer."
B: "Okay, first _____ (check) that your computer is connected to the network."
A: "It is."
B: "Then _____ (open) the web browser and _____ (type) the URL you want to visit."
A: "I've done that but it's not working."
B: "Hmm, _____ (clear) the cache and _____ (restart) the browser."
A: "Alright, let me try that."

Respuestas:

Check
Open, type
Clear, restart

Actividad 1: Completa las órdenes en el imperativo

Completa las siguientes órdenes en el imperativo de forma afirmativa o negativa.

Ejemplo: _____ (sit) down! - Sit down!

_____ (not talk) on the phone while driving. - Don't talk on the phone while driving.
_____ (be) careful when handling sensitive data. - Be careful when handling sensitive data.
_____ (not click) on suspicious links. - Don't click on suspicious links.
_____ (turn off) your computer before leaving. - Turn off your computer before leaving.
_____ (not share) your password with anyone. - Don't share your password with anyone.
Respuestas:

Don't talk
Be
Don't click
Turn off
Don't share
Actividad 2: Ordena las palabras en el imperativo

Ordena las palabras para formar órdenes en el imperativo.

Ejemplo: here / Come - Come here.

a / password / choose / strong - Choose a strong password.
network / your / secure / make - Make your network secure.
your / always / update / software - Always update your software.
personal / not / use / devices - Do not use personal devices.
the / on / click / button / start - Click on the start button.
Respuestas:

Choose a strong password.
Make your network secure.
Always update your software.
Do not use personal devices.
Click on the start button.
Actividad 3: Completa las oraciones con el imperativo negativo

Completa las siguientes oraciones con el imperativo negativo.

Ejemplo: _____ (not open) unknown emails. - Do not open unknown emails.

_____ (not forget) to log out before leaving. - Don't forget to log out before leaving.
_____ (not give) out personal information. - Don't give out personal information.
_____ (not download) unauthorized software. - Don't download unauthorized software.
_____ (not leave) your computer unattended. - Don't leave your computer unattended.
_____ (not write) down passwords. - Don't write down passwords.
Respuestas:

Don't forget
Don't give
Don't download
Don't leave
Don't write

Actividad 1: Instrucciones para desarmar y limpiar una computadora

Instrucciones: Completa las siguientes instrucciones utilizando el imperativo en inglés.

_____ off the computer and unplug it from the power source.
_____ the side panel of the computer case.
_____ a can of compressed air to blow out any dust.
_____ a soft cloth to wipe down any remaining dust.
_____ any loose cables or components.
_____ the side panel back onto the computer case.
_____ the computer back in and turn it on.
Respuestas:

Turn
Remove
Use
Use
Check
Put
Plug
Actividad 2: Instrucciones para instalar un programa en una computadora

Instrucciones: Completa las siguientes instrucciones utilizando el imperativo en inglés.

_____ the program file on your computer.
_____ the installation file.

_____ "Next" to start the installation process.
_____ "Agree" to the terms and conditions.
_____ the installation location.
_____ "Install" to start the installation process.
_____ "Finish" to complete the installation.
Respuestas:

Locate
Run
Click
Click
Select
Click
Click

Handout:

My Daily Routine

Check for updates: _____

Run virus scan: _____

Clean keyboard: _____

Delete old files: _____

Organize files: _____

Back up important data: _____

Clean computer screen: _____

Shut down computer: _____

Example student routine:

I always check for updates in the morning before starting work. I often run a virus scan at night before shutting down my computer. I rarely clean my keyboard, but I try to delete old files and organize my files once a week. I always back up important data once a month. I clean my computer screen every other day.

Handout:

My Daily Routine

Check for updates: _____

Run virus scan: _____

Clean keyboard: _____

Delete old files: _____

Organize files: _____

Back up important data: _____

Clean computer screen: _____

Shut down computer: _____

Example student routine:

I always check for updates in the morning before starting work. I often run a virus scan at night before shutting down my computer. I rarely clean my keyboard, but I try to delete old files and organize my files once a week. I always back up important data once a month. I clean my computer screen every other day.

BRAINSTORMING CUES

Self-introduction → name → age

Place of birth → country → city

education → Field of study → Job / occupation

family → siblings → Marital status → children

personality → hobbies

I'm _____

My friends call me _____

Please call me _____

I go by _____

I'm from _____ , _____ (city, country)

I grew up / was raised in _____. (city, country)

I'm originally from _____ but I've lives in _____ since _____ / I was 21.

I was born in _____, but _____ feels like home to me.

I just turned ____

I'm turning ___ and it's fantastic!

I'm in my early, mid, late ____ / ____ / _____ 's.

If you don't mind, I prefer not to say.

I'm over ____ but I still feel young at heart.

You know, I've lost track of my age.

My major was _____

_____ was my major.

I focused my studies / major in _____

I started with _____ but finished with _____.



I work for _____.

I'm a _____ at _____

I've always wanted to become an _____

I've been preparing to work in the _____

I'm in-between jobs

I'm happily married / single / divorced / separated.

My daughter is _____ years old / in grade school.

I'm single and _____

I'm single and plan to stay single

For now there's nobody to get married to, but _____

TELL US ABOUT YOURSELF IN AS MUCH DETAIL AS POSSIBLE

Hello my name is Kim Tina I'm 26 years old and I'm studying English to become an interpreter in the future, Studying English is very hard but very interesting to me it's really challenging. I worked as reprieve sales representative at as domestic mobile career. I graduated _____--University and studied economics. Studying English is very hard but very interesting to me it's really challenging. As an interpreter, I want to start my career at _____(company) / in _____ (ing)

hello and welcome to lesson one of the business English course essential job vocabulary

let's begin by answering the question where do you work?

I work at

I work in

I work for

and I work with

each preposition is correct in a different context I work at and I work for are used with the name of the company for example

I work at espresso English

I work for Nike

you can also use for if you work directly for a famous person

I work for Tom Cruise I'm his public relations manager

I work in is used in for situations you can say I work in a place

I work in an office

I work in a school

I work in a factory

you can also use in with a city or a country

I work in Paris

I work in France

I work in is also used with a department or area of a company

I work in the marketing department

I work in human resources

I work in sales

finally you can use I work in with a general area

I work in finance

I work in medical research

I work in consulting

you can say **I work with** the things or people that are the objects of your day-to-day work for example

I work with computers I'm a teacher

I work with special needs children

if you want to add more details about your work you can say

I'm responsible for

I'm responsible for updating the company website

I'm responsible for interviewing candidates for jobs

after I'm response before use the ing form of the verb let's review

I work at a company

I work for a company or a person I work in a place city country department or general area

I work with people or things

in conversational English the question where do you work is commonly phrased as

what do you do or what do you do for a living

how do you answer this question if you don't have a job you can say

I'm unemployed or more indirectly I'm between jobs at the moment

if you work for yourself you can say

I'm self-employed

if you have your own company you can say

I own a small business or more specifically I own a restaurant or I own a graphic design company

now let's learn some essential employment vocabulary when you are officially accepted into a new job at a company you are hired by the company for example

I was hired by an insurance company just two weeks after graduating from college

when you're hired you become an **employee** of the company the company becomes your **employer** the other **employees** in the company are your **colleagues** or **coworkers** the person above you who is responsible for your work is your **boss or supervisor** as an employee of the company you **earn** a **salary** money you receive regularly for your work don't make the mistake of saying win a salary or get a salary the correct verb is earn if you're good at your job you might get a **pay raise or a raise** an increase in your salary you could also **get a promotion** an increase in importance authority at the end of the year some companies give their employees a **bonus extra** money for work well done the opposite of hire is **fire** when your company forces you to leave your job for example **Peter was fired because he never came to work on time** usually if someone is fired it's because they did something bad if an employee **loses his or her job** because of a neutral reason like the company reducing its size then we say the employee was laid off for example Donna was laid off when her company started having financial problems if you decide to leave your job there are three verbs you can use **I'm going to quit my job I'm going to leave my job I'm going to resign** quit is informal resign is formal and leave can be formal or informal when an old person decides to stop working the verb for this is **retire** in most countries people retire around age 65 let's review the vocabulary you learned in this lesson

Computer Maintenance Checklist

Use this checklist to keep your computer running smoothly.

1. Keep your computer clean and dust-free. Use a soft, lint-free cloth to wipe down the screen and the keyboard. 2. Keep your computer cool. Make sure there is proper ventilation around the computer and that it is not placed in direct sunlight. 3. Back up important files regularly. Use an external hard drive or cloud storage to keep important files safe. 4. Keep your antivirus software up to date. Run regular scans to check for viruses and malware. 5. Remove unnecessary programs and files. This will free up space on your computer and make it run faster.	6. Update your operating system and other software regularly. This will ensure that your computer is protected from security vulnerabilities and that it runs smoothly. 7. Restart your computer regularly. This will help clear out any temporary files and improve performance. 8. Keep your passwords secure. Use strong passwords and don't share them with anyone. 9. Avoid downloading and installing software from unknown sources. This can lead to security risks and malware infections. 10. Keep your computer physically secure. Lock your computer when you step away from it and don't leave it unattended in public places.

Present Simple, Frequency Adverbs and Prepositions Practice:

1. How often do you _____ (back up / clean / update) your computer?
2. Answer: I back up / clean / update my computer (frequency adverb) once a week / every month / twice a year.

3. Do you use an external hard drive or cloud storage _____ (for / to / with) back up your important files?
4. Answer: I use an external hard drive or cloud storage (preposition) to back up my important files.

5. How do you keep your antivirus software up to date?
6. Answer: I _____ (download / install / update) the latest updates (noun) regularly.

7. Have you ever _____ (downloaded / installed / updated) software from unknown sources?
8. Answer: No, I haven't. I always _____ (download / install / update) software from trusted sources.

9. What should you do if you need to step away from your computer in a public place?
10. Answer: I should lock my computer or _____ (take away / shut down / leave on) it (preposition) until I return.

Activity: Computer Maintenance Checklist Quiz

1. Divide students into pairs or small groups.
2. Hand out the Computer Maintenance Checklist handout.
3. Instruct students to read the handout and familiarize themselves with the checklist.
4. Explain that you will read a series of questions related to the checklist and they must answer them using present simple, frequency adverbs, prepositions, and technical vocabulary.
5. Ask the following questions and have the groups discuss and write down their answers:

- How often should you clean your computer's keyboard? (answer: regularly, once a week)
- What should you use to clean your computer's screen? (answer: a microfiber cloth)
- Where should you store your computer when not in use? (answer: in a dry, cool place)
- How often should you update your antivirus software? (answer: regularly, at least once a week)
- What should you use to clean the inside of your computer? (answer: compressed air)

6. After all questions have been asked, go over the answers as a class and have students explain why they chose their answers.
7. Encourage students to discuss and share any additional tips they have for computer maintenance.
8. Collect the handouts for assessment purposes.

Activity: My Daily Routine

Start by giving each student a handout with a list of activities related to computer maintenance, such as "check for updates," "run virus scan," "clean keyboard," etc. Make sure to include technical vocabulary and prepositions of time (e.g. "in the morning," "at night," etc.).

Have the students use the present simple tense to write out their daily routine for computer maintenance using the provided activities. Encourage the use of frequency adverbs (e.g. "always," "often," "rarely," etc.) to describe how often they perform each activity.

Once students have completed their routine, pair them up with a partner and have them discuss their routines. Encourage students to ask questions and give feedback to each other on their use of grammar and technical vocabulary.

Finally, have a few students share their routines with the class and lead a class discussion on the importance of computer maintenance and the different techniques and tools used in the process.

Example handout:

My Daily Routine

Check for updates: _____
Run virus scan: _____
Clean keyboard: _____
Delete old files: _____
Organize files: _____
Back up important data: _____
Clean computer screen: _____
Shut down computer: _____

Example student routine:

I always check for updates in the morning before starting work. I often run a virus scan at night before shutting down my computer. I rarely clean my keyboard, but I try to delete old files and organize my files once a week. I always back up important data once a month. I clean my computer screen every other day.

the use of adjectives, comparative, and superlative adjectives in context, with worksheet

Sure, here's an example worksheet:

Title: Describing Tech Products

Instructions:

1. Look at the tech products listed below and complete the chart with the adjectives that best describe each product.
2. Use comparative and superlative adjectives to compare and contrast the tech products.

Tech Products:

Laptop
Tablet
Smartphone
Smartwatch

Adjectives:

Durable
Portable
Lightweight
Efficient
User-friendly
High-quality
Versatile
Innovative

Chart:

Tech Product	Adjective	Comparative	Superlative
Laptop			
Tablet			
Smartphone			
Smartwatch			

Example Answers:

Tech Product	Adjective	Comparative	Superlative
Laptop	Durable	More durable	Most durable
	High-quality	Higher-quality	Highest-quality
	Efficient	More efficient	Most efficient
	Versatile	More versatile	Most versatile
	Portable	More portable	Most portable
	User-friendly	More user-friendly	Most user-friendly
	Innovative	More innovative	Most innovative

Tech Product	Adjective	Comparative	Superlative
	Lightweight	Lighter-weight	Lightest-weight
Tablet	Portable	More portable	Most portable
	User-friendly	More user-friendly	Most user-friendly
	Versatile	More versatile	Most versatile
	Innovative	More innovative	Most innovative
	Lightweight	Lighter-weight	Lightest-weight
Smartphone	User-friendly	More user-friendly	Most user-friendly
	Efficient	More efficient	Most efficient
	Innovative	More innovative	Most innovative
	High-quality	Higher-quality	Highest-quality
Smartwatch	Lightweight	Lighter-weight	Lightest-weight
	User-friendly	More user-friendly	Most user-friendly
	Innovative	More innovative	Most innovative
	Versatile	More versatile	Most versatile
	Portable	More portable	Most portable

example related to computer specifications and features, with worksheet

Sure, here's an example lesson plan with a worksheet related to computer specifications and features:

Title: Understanding Computer Specifications and Features

Level: Basic

Objectives:

To introduce and familiarize students with technical terms related to computer specifications and features
To provide practice in reading and interpreting computer specifications and features
To develop the ability to compare and contrast different computer models based on their specifications and features
Materials:

Computer specifications and features worksheet (see attached)

Computer advertisements or product pages for different models
Whiteboard and markers
Procedure:

Warm-up (10 minutes):
Start by asking the students if they own a computer and what kind of computer they have. Ask them to describe the computer's specifications and features. Write some of the technical terms they use on the whiteboard.

Presentation (20 minutes):
Introduce the students to different technical terms related to computer specifications and features, such as CPU, RAM, storage, operating system, display, resolution, graphics card, etc. Use the whiteboard to write down each term and its definition. Use visuals and examples to help students understand each term.

Practice (30 minutes):
Distribute the computer specifications and features worksheet to the students. Ask them to read each computer model's specifications and features carefully and answer the questions on the worksheet. After they finish, ask them to compare and contrast the different models based on their specifications and features.

Follow-up (15 minutes):
Divide the class into small groups and distribute different computer advertisements or product pages for different models. Ask the students to read each advertisement or product page and identify the technical terms related to computer specifications and features. Have each group present their findings to the class.

Conclusion (5 minutes):
Summarize the lesson by reviewing the technical terms related to computer specifications and features that the students learned. Ask them to reflect on what they learned and how they can apply their knowledge in the future.

Computer specifications and features worksheet:

Instructions: Read each computer model's specifications and features carefully and answer the questions below.

Computer Model A:
CPU: Intel Core i5
RAM: 8GB DDR4
Storage: 256GB SSD
Operating System: Windows 10
Display: 14-inch HD LED-backlit
Graphics Card: Intel UHD Graphics

What kind of CPU does Computer Model A have?

How much RAM does Computer Model A have?
What kind of storage does Computer Model A have?
What operating system does Computer Model A use?
What is the size and type of Computer Model A's display?
What kind of graphics card does Computer Model A have?

Computer Model B:
CPU: AMD Ryzen 3
RAM: 4GB DDR4
Storage: 1TB HDD
Operating System: Windows 10 Home
Display: 15.6-inch Full HD LED-backlit
Graphics Card: AMD Radeon Graphics

What kind of CPU does Computer Model B have?
How much RAM does Computer Model B have?
What kind of storage does Computer Model B have?
What operating system does Computer Model B use?
What is the size and type of Computer Model B's display?
What kind of graphics card does Computer Model B have?

DDR4 stands for "Doble Tasa de Datos de Cuarta Generación" in Spanish.

example ofthe use of adjectives, comparative, and superlative adjectives in context, giving examples of each.

Sure! Here is an example:

Context: A company is looking for a new computer to buy for their employees.

Adjectives: fast, reliable, efficient, powerful Comparative adjectives: faster, more reliable, more efficient, more powerful Superlative adjectives: fastest, most reliable, most efficient, most powerful

Example sentences:

- The new computer is fast, but the older model is even faster.
- A reliable computer is important for our work, but we need something more reliable than what we have now.
- We need an efficient computer that can handle multiple tasks at once.
- This computer is powerful, but the most powerful one we have seen so far is too expensive.

In this example, the technical vocabulary could be related to computer specifications and features, such as RAM, processing speed, and storage capacity. Students could practice using these adjectives, comparatives, and superlatives to describe and compare different computers, and then make recommendations based on their findings.

Sure! Here's an example activity:

Title: Correct the Mistakes

Objective: To practice using adjectives, comparative, and superlative adjectives in context by identifying and correcting mistakes.

Materials:

- Handout with sentences containing mistakes related to technical topics
- Worksheet with the same sentences, but with blanks where the mistakes are

Procedure:

1. Introduce the activity by reviewing adjectives, comparative, and superlative adjectives.
2. Distribute the handout to the students and ask them to read each sentence carefully.
3. Instruct the students to identify the mistakes in each sentence related to adjectives, comparative, and superlative adjectives.
4. Once the students have identified the mistakes, distribute the worksheet with the same sentences, but with blanks where the mistakes are. Instruct the students to write the correct form of the adjective, comparative or superlative adjective in each blank.
5. Allow the students to work on the worksheet individually or in pairs.
6. Review the answers as a class and discuss any questions the students may have.

Example sentences:

1. My laptop is more fast than my desktop computer.
2. The new printer is the more faster than the old one.
3. This software is easier to use than the previous version.
4. The camera has a better quality picture than the other one.
5. The internet connection in the office is more slow than at home.

Worksheet:

1. My laptop is _____ than my desktop computer.
2. The new printer is _____ than the old one.
3. This software is _____ to use than the previous version.
4. The camera has a _____ quality picture than the other one.
5. The internet connection in the office is _____ than at home.

Correct answers:

1. faster
2. faster
3. easier
4. better
5. slower

computer model's specifications and features
advertisement or product page and identify the technical terms related to computer specifications and features

Test your knowledge of the vocabulary and grammar.

Full name_____
Program ID _____

Choose

1. The synonym for **"distorted"** is:
a) swamped
b) garbled
c) outlined

2. The synonym of **"easily"** is:
a) constrained
b) upfront
c) go smootly

3. they manage to fix all the bugs before the _____
a) deadline
b) smoothly
c) roughly

4.The opposite of **"nontechnical"** is:
a) technical
b) outline
c) postpone

5.There's a good rule of _____ you can follow in this case.
a) toe
b) thumb
c) finger

6. you must be able to _____ on URLs
a) delete
b) touch
c) click

7.Choose the option where **"ch"** is NOT pronounced as "k":
a) technical
b) attach
c) archive

Choose the verb in the correct time (past, present perfect or passive voice)

8. I_____ you yesterday about the bug.
a) told
b) was told
c) have be told

9. An error _____ .
a) has occurred
b) was occurred
c) has been occurred

10. the code _____ to fire him.
a) have been tested
b) was tested
c) tested

11. The rest of the codes _____ soon.
a) will be finished
b) will finished
c) is finishing

Match the synonyms.

1. Fix ( )	a. Put off
2. Setback ( )	b. Repair
3. Pitfall ( )	c. distorted
4. Face ( )	d. Deal with
5. Postpone ( )	e. More than
6. Replicate ( )	f. Difficult
7. Upfront ( )	g. Reproduce
8. Over ( )	h. In advance
9. Swamped ( )	i. unseen danger
10. Garbled ( )	j. inundated

Write down the meaning of the following phrases:

1) A lot of setbacks have happened during the process.

_____

_____

2) Some workmates were transferred to a different department.

_____

.

3. If you are swamped with requests and bug reports, keeping a strategy that can help you store and prioritize all the incoming messages. Keep all of your communication lines organized

_____
_____
_____
_____
_____
_____
_____
_____