



MICROCREDENZIALI PER LA CREAZIONE DI CONTENUTI DIGITALI

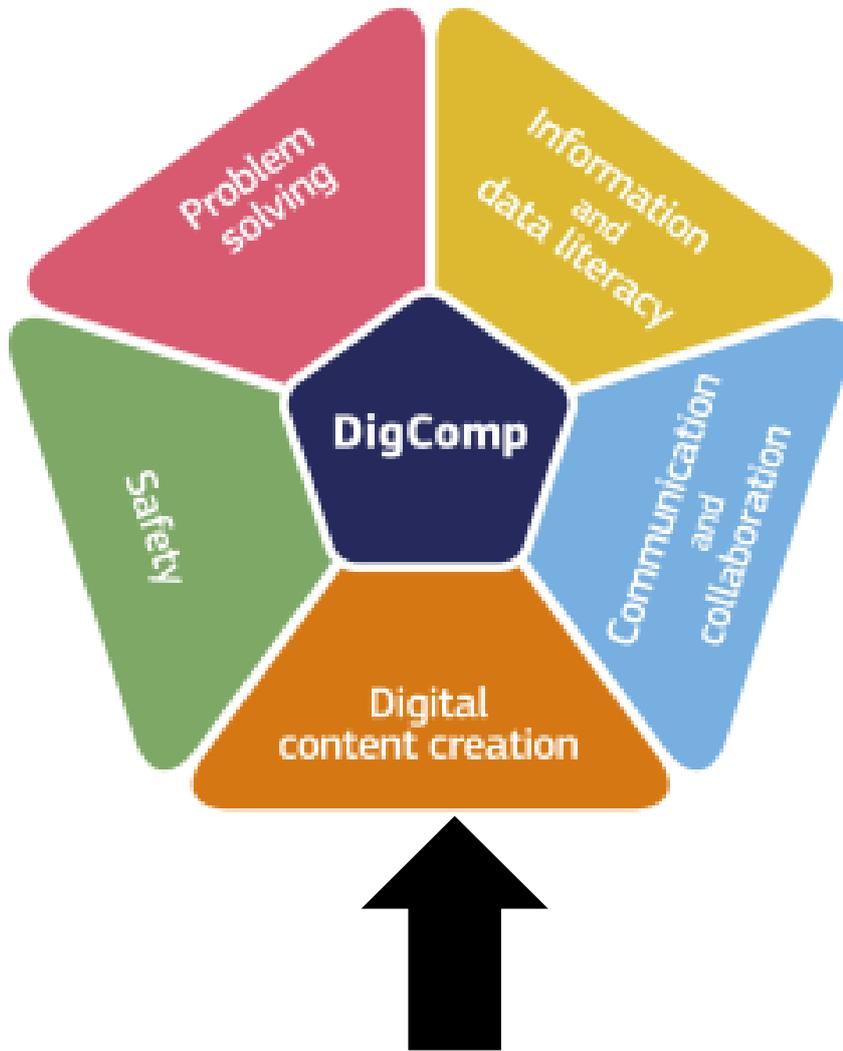
Competenza 3.4: PROGRAMMAZIONE

DSW
DIGITAL SKILLS WALLET



**Cofinanziato
dall'Unione europea**

Finanziato dall'Unione europea. Le opinioni espresse appartengono, tuttavia, al solo o ai soli autori e non riflettono necessariamente le opinioni dell'Unione europea o dell'Agencia esecutiva europea per l'istruzione e la cultura (EACEA). Né l'Unione europea né l'EACEA possono esserne ritenute responsabili.



Contenuto

Contenuto.....	3
.....	5
LIVELLO BASE.....	5
Linguaggi di programmazione - tipi e classificazione (MC 3.4.A.1)	5
IDE - Ambiente di sviluppo integrato (MC 3.4.A.2)	8
Scegliere e installare un IDE (MC 3.4.A.3)	10
Documentazione di programmazione (MC 3.4.A.4).....	12
Domini di applicazione (MC 3.4.A.5)	14
.....	16
LIVELLO INTERMEDIO	16
Specifiche/funzionalità di un'applicazione (MC 3.4.B.1)	16
Scegliere un linguaggio di programmazione corrispondente alla piattaforma di destinazione (MC 3.4.B.2) ..	19
Scegliere un IDE corrispondente al linguaggio di programmazione (MC 3.4.B.3).....	21
Sintassi del linguaggio di programmazione (MC 3.4.B.4)	23
Tipi di dati (MC 3.4.B.5)	25
Variabili e operatori (MC 3.4.B.6).....	27
Dichiarazioni sul flusso di controllo (MC 3.4.B.7)	29
Eeguire il file sorgente (MC 3.4.B.8)	31
.....	33
LIVELLO AVANZATO	33
Metodi/funzioni (MC 3.4.C.1).....	33
Operazioni di ingresso/uscita (I/O) (MC 3.4.C.2).....	36
Librerie e moduli (MC 3.4.C.3).....	38
Paradigmi di programmazione (MC 3.4.C.4)	40
Programmazione imperativa e orientata agli oggetti (MC 3.4.C.5)	42
Programmazione logica e funzionale (MC 3.4.C.6)	44
Programmazione concorrente (MC 3.4.C.7).....	46
Test e debug (MC 3.4.C.8)	48
Linguaggi di programmazione compilati e interpretati (MC 3.4.C.9).....	50
Linguaggi di markup (MC 3.4.C.10).....	52
.....	54
LIVELLO ESPERTO.....	54



Progettare la soluzione di un problema complesso (MC 3.4.D.1).....	54
Project management (MC 3.4.D.2).....	57
Leadership. Creatività (MC 3.4.D.3)	59
APPENDICE I RISULTATI DI APPRENDIMENTO PER L'AREA DI COMPETENZA: CREAZIONE DI CONTENUTI DIGITALI	61
COMPETENZA: 3.4 Programmazione.....	61
INTRODUZIONE:.....	62
PREREQUISITI.....	63

LIVELLO BASE
(Livello 1 e Livello 2)



Linguaggi di programmazione - tipi e classificazione (MC 3.4.A.1)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Linguaggi di programmazione: tipi e classificazione Codice: MC 3.4.A.1
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	BASE
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.A.1)

Risultati di apprendimento (rif. LO di livello 1-2 3.4.1 - 3.2.4):

- Comprendere i tipi di applicazione che possono essere sviluppati in un linguaggio di programmazione.
- Spiegare le caratteristiche dei diversi tipi di applicazioni per quanto riguarda la piattaforma e l'ambiente di lavoro.
- È consapevole di come il linguaggio di programmazione possa essere utilizzato per sviluppare applicazioni per vari sistemi e ambienti.
- Enumerare un elenco con un linguaggio di programmazione comune.

Descrizione (MC 3.4.A.1)

A questo livello fondamentale di competenza nella programmazione nell'ambito della creazione di contenuti digitali, gli sviluppatori devono comprendere i diversi tipi di applicazioni sviluppate attraverso i linguaggi di programmazione. Devono avere una conoscenza di base del fatto che le applicazioni desktop vengono eseguite su sistemi operativi come Windows, MacOS o Linux, offrendo funzionalità autonome. Inoltre, devono capire che le applicazioni mobili sono destinate a piattaforme come Android e iOS, adattandosi ai dispositivi portatili. Devono sapere che le applicazioni web sono utilizzate per la creazione di siti web e servizi, accessibili attraverso vari browser. Devono sapere che le applicazioni embedded comportano una programmazione specializzata per funzioni dedicate in dispositivi come i sistemi IoT, garantendo la compatibilità con ambienti informatici specifici.

Gli sviluppatori devono enumerare le caratteristiche dei diversi tipi di applicazione, riconoscendo le caratteristiche chiave dei tipi di applicazione in base alla loro piattaforma di lavoro (desktop, mobile, web, embedded) e all'ambiente (software standalone, ottimizzato per interfacce touch, accesso tramite Internet, mirato a funzioni specifiche in dispositivi che richiedono un utilizzo efficiente delle risorse e l'elaborazione in tempo reale), riconoscendo la differenza tra linguaggi di programmazione specifici per la piattaforma e linguaggi di programmazione multipiattaforma.

A livello di base, gli individui dovrebbero avere familiarità con esempi di linguaggi di programmazione, tra cui C, C++, C#, Java, Kotlin, JavaScript, HTML, XML, React, Python, Ruby, Prolog, Lisp e Scala.

Domande (MC 3.4.A.1)

1. Può elencare i principali tipi di applicazioni in base alla loro piattaforma di implementazione (desktop, mobile, web, embedded)?
2. Descrivete brevemente le differenze tra le modalità di sviluppo delle applicazioni per i vari sistemi e ambienti.
3. Quali sono le caratteristiche principali delle applicazioni desktop, in particolare di quelle eseguite su sistemi operativi come Windows, MacOS o Linux?
4. Spiegare le caratteristiche delle applicazioni mobili progettate per piattaforme come Android e iOS.
5. Descrivere come i linguaggi di programmazione possono essere utilizzati per creare applicazioni web, siti e servizi web.
6. Come si può utilizzare efficacemente un linguaggio di programmazione nella programmazione di sistemi embedded?
7. Fornire esempi di linguaggi di programmazione per applicazioni desktop/mobile/web/embedded.

IDE - Ambiente di sviluppo integrato (MC 3.4.A.2)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	IDE - Ambiente di sviluppo integrato Codice: MC 3.4.A.2
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	BASE
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.A.2)

Risultati di apprendimento (rif. LO di livello 1-2 3.4.5 - 3.2.6):

- Identificare che cos'è un IDE - Integrated Development Environment (ambiente di sviluppo integrato).
- È consapevole delle principali caratteristiche e funzionalità dei diversi IDE.

Descrizione (MC 3.4.A.2)

Al livello fondamentale di competenza nella programmazione, gli individui dovrebbero riconoscere l'IDE come acronimo di "Integrated Development Environment" (ambiente di sviluppo integrato) e comprenderne il ruolo nelle attività di programmazione relative alla creazione di contenuti digitali.

Gli sviluppatori devono essere consapevoli delle principali caratteristiche e funzionalità dei diversi IDE. Comprendono i diversi strumenti e le capacità offerte dai vari IDE, dimostrando la capacità di navigare e utilizzare questi strumenti in modo efficace. Questa consapevolezza include la familiarità con le funzionalità adatte alla creazione di contenuti digitali, garantendo l'efficienza delle attività di codifica, debug e test. Le caratteristiche principali che si trovano comunemente in un IDE sono: editor di codice, compilatore/interprete, debugger, Project management, controllo della versione, automazione della compilazione, modelli e snippet di codice, refactoring del codice, analisi del codice per potenziali errori, strumenti di completamento automatico, evidenziazione di errori e avvertenze, strumenti di ricerca e sostituzione, navigazione nel codice, integrazione della documentazione o dei riferimenti API esterni, integrazione dei test unitari, personalizzazione ed estensioni.

Domande (MC 3.4.A.2)

1. Cosa significa IDE e come lo definirebbe?
2. Spiegare il ruolo dell'IDE nel facilitare lo sviluppo del software.
3. Potete identificare e descrivere brevemente tre caratteristiche principali che si trovano comunemente in un ambiente di sviluppo integrato (IDE) e che contribuiscono a migliorare la produttività e l'esperienza di codifica di uno sviluppatore?
4. Quali strumenti e funzionalità specifiche fornisce agli sviluppatori per identificare e risolvere i bug durante il processo di esecuzione del codice?
5. In che modo il supporto integrato per i sistemi di controllo delle versioni, come Git, avvantaggia gli sviluppatori nella gestione delle modifiche al codice e nella collaborazione con gli altri?
6. In che modo la funzionalità di completamento automatico in un IDE contribuisce all'efficienza della codifica e alla riduzione degli errori?

Scegliere e installare un IDE (MC 3.4.A.3)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Scegliere e installare un IDE. Codice: MC 3.4.A.3
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	BASE
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.A.3)

Risultati di apprendimento (rif. LO di livello 1-2 3.4.7 - 3.2.8):

- Trovate e scegliete un IDE corrispondente a un linguaggio di programmazione.
- Installare un IDE.

Descrizione (MC 3.4.A.3)

Al livello base di competenza nella programmazione per la creazione di contenuti digitali, i programmatori devono possedere competenze pratiche relative alla scelta e all'utilizzo di ambienti di sviluppo integrati (IDE).

In primo luogo, devono dimostrare la capacità di trovare e scegliere un IDE adatto a uno specifico linguaggio di programmazione, considerando fattori come il supporto del linguaggio, le caratteristiche e la compatibilità. Questo riflette una comprensione fondamentale dell'importanza degli IDE nel processo di sviluppo.

In secondo luogo, devono dimostrare di essere abili nell'installazione di un IDE, mostrando l'abilità pratica di impostare l'ambiente di sviluppo scelto. Questo include la navigazione nei processi di installazione, la configurazione delle impostazioni, la garanzia di una configurazione continua e la gestione degli aggiornamenti. Ciò potrebbe comportare la configurazione degli strumenti di compilazione, l'impostazione delle opzioni di compilazione e la comprensione del processo di compilazione.

Domande (MC 3.4.A.3)

1. Come si fa a trovare su Internet un elenco degli ambienti di sviluppo integrato (IDE) disponibili per uno specifico linguaggio di programmazione?
2. Dato un insieme di caratteristiche e funzionalità associate a diversi IDE, potete spiegare il vostro processo decisionale nella scelta di un IDE per un particolare linguaggio di programmazione?
3. Evidenziate almeno tre caratteristiche chiave a cui dareste priorità nella scelta di un IDE.
4. Descrivere le fasi di installazione di un IDE con impostazioni predefinite o personalizzate. Includere considerazioni quali il download e l'esecuzione di un programma di installazione, la configurazione delle impostazioni e la gestione degli aggiornamenti.
5. Come si configurano gli strumenti di compilazione, si impostano le opzioni di compilazione e si comprende il processo di compilazione all'interno di un IDE?
6. Quando si installa un IDE con impostazioni personalizzate, come si può adattare la configurazione per soddisfare requisiti specifici?

Documentazione di programmazione (MC 3.4.A.4)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Documentazione di programmazione Codice: MC 3.2.A.4
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	BASE
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.A.4)

Risultati di apprendimento (rif. LO di livello 1-2 3.4.9 - 3.2.11):

- Spiegare e trovare la documentazione di programmazione del linguaggio.
- È in grado di eseguire un semplice programma.

Descrizione (MC 3.4.A.4)

Al livello base di competenza nella programmazione per la creazione di contenuti digitali, gli sviluppatori devono dimostrare di saper comprendere e reperire la documentazione del linguaggio di programmazione. Devono riconoscere quattro tipi fondamentali: la documentazione ufficiale del linguaggio, i tutorial, i post/discussioni e i materiali di riferimento. Queste conoscenze fondamentali sottolineano l'importanza di risorse diverse per un apprendimento e una risoluzione dei problemi efficaci.

In secondo luogo, le persone devono mostrare abilità pratiche nell'esecuzione di un semplice programma ispirato alla documentazione. Devono creare, modificare e salvare file di codice sorgente in un ambiente di sviluppo integrato (IDE), comprendendo le estensioni specifiche dei file. Inoltre, sono in grado di costruire e compilare il codice, spesso facendo riferimento alla documentazione ufficiale come guida. Queste competenze riflettono un approccio pratico alla codifica in un contesto di creazione di contenuti digitali, gettando le basi per attività di programmazione più avanzate.

Domande (MC 3.4.A.4)

1. Spiegare i quattro tipi di documentazione di un linguaggio di programmazione.
2. Descrivere il processo di creazione, modifica e salvataggio dei file di codice sorgente in un ambiente di sviluppo integrato (IDE).
3. Come ci si assicura che i file abbiano le estensioni corrette per il linguaggio di programmazione con cui si lavora?
4. In che modo si sfrutta la documentazione del linguaggio di programmazione, in particolare quella ufficiale, quando si lavora a un progetto di codifica?
5. Se si verifica un errore durante il processo di creazione e compilazione del codice, come si affronta la risoluzione del problema?

Domini di applicazione (MC 3.4.A.5)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Domini di applicazione. Codice: MC 3.2.A.5
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	BASE
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.A.5)

Risultati di apprendimento (rif. LO di livello 1-2 3.4.12 - 3.2.13):

- Classificare i domini di applicazione.
- Identificare i requisiti e i vincoli specifici per un determinato dominio applicativo.

Descrizione (MC 3.4.A.5)

Al livello base di competenza nella programmazione per la creazione di contenuti digitali, gli sviluppatori devono essere in grado di classificare i domini applicativi, intendendoli come aree di vita reale in cui le applicazioni sono destinate ad essere applicate. Questa conoscenza di base garantisce la consapevolezza dei diversi contesti in cui operano le applicazioni per la creazione di contenuti digitali, come la finanza, la sanità, l'istruzione, il commercio elettronico, i trasporti e la logistica, l'intrattenimento e i media, la produzione e l'automazione industriale, il governo, i servizi pubblici, l'IoT (Internet of Things), le scienze.

In secondo luogo, i programmatori dimostrano la capacità di identificare richieste e vincoli specifici all'interno di un determinato dominio applicativo. Sono in grado di porre domande ponderate su considerazioni critiche come i requisiti degli utenti, le strutture dei dati, gli algoritmi, le misure di sicurezza e gli standard di conformità. Questa abilità riflette la capacità di analizzare e comprendere le sfide uniche associate alla programmazione in un contesto specifico, ponendo le basi per una creazione di contenuti digitali efficace e mirata.

Domande (MC 3.4.A.5)

1. Definire il termine "dominio applicativo" (o dominio problematico) nel contesto dello sviluppo del software.
2. Fornite cinque esempi di diversi domini applicativi nello sviluppo del software.
3. In che modo il riconoscimento dell'ambito e del contesto di un dominio applicativo influenza la progettazione e l'implementazione del software?
4. Spiegare il ruolo del dominio applicativo nel guidare gli sviluppatori durante il processo di sviluppo del software.
5. Elencare e descrivere brevemente le considerazioni chiave che uno sviluppatore dovrebbe fare quando lavora in un dominio applicativo specifico.
6. Fornire esempi di applicazioni software in almeno tre diversi ambiti applicativi.

LIVELLO INTERMEDIO

(Livello 3 e Livello 4)



Specifiche/funzionalità di un'applicazione (MC 3.4.B.1)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Specifiche/funzionalità di un'applicazione Codice: MC 3.2.B.1
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	INTERMEDIO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.B.1)

Risultati di apprendimento (rif. LO di livello 3-4 3.4.14 - 3.2.15):

- Spiegare le principali specifiche/funzionalità di un'applicazione in un particolare dominio.
- Stabilire le principali specifiche/funzionalità di un'applicazione in un particolare dominio.

Descrizione (MC 3.4.B.1)

Al livello intermedio di competenza nella programmazione per la creazione di contenuti digitali, gli sviluppatori devono essere in grado di spiegare le principali specifiche e funzionalità richieste per un'applicazione all'interno di un dominio specifico. Ciò implica la comprensione del fatto che le specifiche del dominio sono i requisiti unici e le capacità essenziali per rispondere efficacemente alle esigenze di quel dominio. Ad esempio, un'applicazione di e-commerce richiede la registrazione dell'utente, un catalogo di prodotti e un'elaborazione sicura dei pagamenti.

In secondo luogo, le persone devono dimostrare la capacità di stabilire queste specifiche ponendo domande critiche sulle funzioni e sui vincoli futuri. Possono informarsi sui dettagli di registrazione degli utenti, sulle opzioni di ripetizione dei quiz e sui meccanismi di feedback, assicurando una comprensione completa delle funzionalità dell'applicazione. Questo set di competenze fondamentali consente agli sviluppatori di allineare la creazione di contenuti digitali con le caratteristiche e le esigenze distintive di diversi settori applicativi, quali finanza, sanità, istruzione, e-commerce, trasporti e logistica, intrattenimento e media, produzione e automazione industriale, governo, servizi pubblici, IoT (Internet of Things), scienze.

Domande (MC 3.4.B.1)

1. In che modo le specifiche differiscono tra i vari domini applicativi e perché è fondamentale che le funzionalità si allineino alle caratteristiche del dominio?
2. Fornire esempi di specifiche e funzionalità per un'applicazione educativa.
3. Spiegare l'importanza di porre domande critiche sulle funzioni e sui vincoli futuri quando si stabiliscono le specifiche di un'applicazione.
4. Quando si stabiliscono le funzionalità per la registrazione degli utenti in un'applicazione di e-commerce, quali dettagli critici devono essere considerati? Discutete se sia sufficiente chiedere il nome completo, l'età, il telefono o solo l'e-mail e perché.
5. Quando si progettano le funzionalità di un quiz, quali considerazioni si devono fare per quanto riguarda la ripetizione dei quiz? Discutete il processo decisionale per rimescolare le domande e/o le risposte e per fornire feedback o risposte corrette in base alla possibilità di ripetere i quiz.

Scegliere un linguaggio di programmazione corrispondente alla piattaforma di destinazione (MC 3.4.B.2)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Scegliere un linguaggio di programmazione corrispondente alla piattaforma di destinazione. Codice: MC 3.2.B.2
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	INTERMEDIO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.B.2)

Risultati di apprendimento (rif. LO di livello 3-4 3.4.16 - 3.2.17):

- È consapevole dei criteri di scelta di un linguaggio di programmazione corrispondente alla piattaforma di destinazione.
- Scegliere un linguaggio di programmazione specifico corrispondente alla piattaforma di destinazione.

Descrizione (MC 3.4.B.2)

Al livello intermedio di competenza nella programmazione, i programmatori devono dimostrare di essere consapevoli dei criteri di scelta di un linguaggio di programmazione in linea con la piattaforma di destinazione. Riconoscendo che i diversi linguaggi eccellono in diversi ambiti applicativi, gli sviluppatori considerano fattori quali il fatto che l'applicazione sia un'applicazione web o mobile, un software desktop, un gioco o uno strumento di analisi dei dati. Inoltre, sono consapevoli dei punti di forza e di debolezza dei linguaggi di programmazione in relazione alla piattaforma scelta, come Java/Kotlin per Android o Swift per iOS.

In secondo luogo, gli sviluppatori scelgono attivamente un linguaggio di programmazione specifico sulla base di una ricerca approfondita dei vincoli dell'applicazione, del pubblico di destinazione e delle piattaforme utilizzate. Considerano gli aspetti finanziari, distinguendo tra opzioni gratuite e commerciali, e valutano la frequenza degli aggiornamenti sia per i linguaggi di programmazione che per gli ambienti di sviluppo integrati.

Domande (MC 3.4.B.2)

1. In che modo fattori come i punti di forza, i punti deboli e le aree di competenza dei linguaggi di programmazione contribuiscono a prendere una decisione consapevole in base alla piattaforma di destinazione?
2. Fornire esempi di linguaggi di programmazione adatti a piattaforme specifiche.
3. Quando si sceglie un linguaggio di programmazione corrispondente alla piattaforma di destinazione, quali sono i fattori da considerare per quanto riguarda i vincoli dell'applicazione, il pubblico di destinazione e le piattaforme utilizzate?
4. Descrivete i passi che fareste per cercare informazioni sui vincoli applicativi relativi al gruppo target e alle piattaforme utilizzate.
5. Quando si sceglie un linguaggio di programmazione specifico, come si valutano gli aspetti finanziari, tra cui il fatto che sia gratuito o commerciale e la frequenza degli aggiornamenti?

Scegliere un IDE corrispondente al linguaggio di programmazione (MC 3.4.B.3)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Scegliete un IDE corrispondente al linguaggio di programmazione. Codice: MC 3.2.B.3
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	INTERMEDIO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.B.3)

Risultati di apprendimento (rif. LO di livello 3-4 3.4.18 - 3.2.19):

- È consapevole dei criteri di scelta di un IDE.
- Scegliere l'IDE giusto per un linguaggio di programmazione specifico.

Descrizione (MC 3.4.B.3)

Al livello intermedio gli sviluppatori dovrebbero conoscere i diversi criteri di selezione di un IDE, considerando fattori quali l'associazione con un linguaggio di programmazione, la compatibilità con la piattaforma, il supporto alla programmazione, gli strumenti integrati e il supporto della comunità. La loro consapevolezza si estende all'importanza dell'integrazione con strumenti esterni e alla considerazione degli ecosistemi IDE. Gli individui comprendono che le preferenze personali giocano un ruolo cruciale nel processo di selezione e sostengono la necessità di testare diversi IDE per determinare il comfort e la produttività.

Inoltre, a questo livello, gli sviluppatori sono in grado di scegliere l'IDE giusto per uno specifico linguaggio di programmazione, allineando la scelta alle esigenze del progetto e alle preferenze personali o del team. Dimostrano di avere dimestichezza con gli IDE più diffusi per vari linguaggi, dimostrando una comprensione sfumata della selezione degli strumenti più adatti per diversi progetti applicativi.

Domande (MC 3.4.B.3)

1. Enumerare almeno tre criteri per la scelta di un ambiente di sviluppo integrato (IDE) quando si intraprende un nuovo progetto di programmazione.
2. Spiegare l'importanza di verificare la compatibilità di un IDE con strumenti o framework esterni. Fornite un esempio di uno scenario in cui la perfetta integrazione con strumenti esterni è fondamentale per migliorare il flusso di lavoro dello sviluppo.
3. Descrivete il concetto di ecosistema IDE e come può essere utile agli sviluppatori.
4. In che modo le preferenze individuali possono influire sulla produttività e sul livello di comfort di uno sviluppatore e perché è consigliabile testare diversi IDE prima di fare una scelta definitiva?
5. Selezionate un linguaggio di programmazione dall'elenco (Java, Python, JavaScript, C++, C#, Swift) e consigliate gli IDE più adatti in base alla vostra comprensione delle loro caratteristiche e capacità. Spiegate le motivazioni delle vostre scelte, tenendo conto dei requisiti del linguaggio e delle preferenze del team.

Sintassi del linguaggio di programmazione (MC 3.4.B.4)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Sintassi del linguaggio di programmazione. Codice: MC 3.2.B.4
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 6 - Massimo 8 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	INTERMEDIO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.B.4)

Risultati di apprendimento (rif. Livello 3-4 LO 3.4.20 - 3.2.22):

- È consapevole della definizione della sintassi di un linguaggio di programmazione.
- È consapevole del ruolo della sintassi di un linguaggio di programmazione.
- È consapevole delle principali regole della sintassi di una lingua.

Descrizione (MC 3.4.B.4)

Al livello intermedio di competenza di programmazione nella creazione di contenuti digitali, gli sviluppatori possiedono conoscenze avanzate sulla sintassi dei linguaggi di programmazione. Sono consapevoli che la sintassi comporta un insieme di regole che governano la struttura e la grammatica del codice, fondamentali per la comprensione del compilatore o dell'interprete. Riconoscendo i sette ruoli della sintassi, gli sviluppatori ne comprendono l'importanza nel parsing, nel rilevamento degli errori, nella leggibilità del codice, nella coerenza del linguaggio, nella definizione del comportamento, nel riflettere l'evoluzione del linguaggio e nel garantire la compatibilità con strumenti e framework.

Inoltre, i programmatori conoscono le principali regole che governano la sintassi di un linguaggio. Ciò include la comprensione di aspetti come i commenti per le note esplicative, i terminatori e i separatori, le parole chiave con significati specifici, le regole sugli identificatori per nominare variabili e funzioni, le regole sugli operatori per eseguire operazioni, le dichiarazioni e le espressioni per le istruzioni, i blocchi di codice definiti attraverso l'indentazione o le parentesi graffe e il significato della sensibilità alle maiuscole. Questa maggiore consapevolezza consente agli sviluppatori di scrivere codice coerente e privo di errori, favorendo la leggibilità, la coerenza e il rispetto delle migliori pratiche nelle applicazioni di programmazione.

Domande (MC 3.4.B.4)

1. Che cosa significa il termine "sintassi" nel contesto di un linguaggio di programmazione?
2. Come descriverebbe il ruolo della sintassi nel garantire la validità e la comprensibilità del codice da parte di un compilatore o di un interprete?
3. Spiegare il ruolo della sintassi nell'analisi del codice sorgente e il suo significato nel processo di compilazione.
4. In che modo la sintassi contribuisce al rilevamento degli errori nella programmazione?
5. In che modo una sintassi ben definita garantisce la coerenza del linguaggio?
6. In che modo le regole del linguaggio, comprese nella sintassi, influenzano il comportamento e la semantica di un'applicazione sviluppata?
7. Fornire esempi di come le regole di sintassi si allineino alle best practice e aiutino a evitare ambiguità nel codice.
8. Descrivere lo scopo dei commenti in un linguaggio di programmazione e il loro ruolo nell'interpretazione del codice.
9. In che modo i terminatori e i separatori contribuiscono alla chiarezza e all'organizzazione del codice, soprattutto nelle dichiarazioni o negli elenchi complessi?

Tipi di dati (MC 3.4.B.5)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Tipi di dati. Codice: MC 3.2.B.5
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 4 - Massimo 6 ore
Livello dell'esperienza di apprendimento che porta alla microcredenziale	INTERMEDIO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.B.5)

Risultati di apprendimento (rif. LO di livello 3-4 3.4.23 - 3.2.24):

- È consapevole della classificazione dei tipi di dati e delle loro caratteristiche.
- Elencare i principali tipi di dati e le loro caratteristiche in un programma linguistico.

Descrizione (MC 3.4.B.5)

Al livello intermedio di competenza nella programmazione, gli sviluppatori devono riconoscere che i tipi di dati classificano le variabili o le espressioni in base ai loro valori, determinando le operazioni consentite, gli intervalli di valori e i requisiti di memoria. Comprendono il ruolo fondamentale dei tipi di dati nell'utilizzo efficiente della memoria, nell'assicurare l'accuratezza dei dati e nell'influenzare la funzionalità complessiva del programma. I programmatori sanno che i tipi di dati disponibili in un linguaggio di programmazione hanno un impatto diretto sulle operazioni applicabili ai dati e modellano il comportamento del programma.

Inoltre, gli sviluppatori dimostrano la capacità di enumerare i principali tipi di dati, spiegandone le rappresentazioni e distinguendo tra forme interne ed esterne. Forniscono esempi, come Integer per i numeri interi, Floating-Point per i numeri reali, Character per i singoli caratteri, String per le sequenze di testo, Boolean per le decisioni binarie, Array per le collezioni organizzate, Pointer per gli indirizzi di memoria e Struct/Record per i tipi composti definiti dall'utente. Inoltre, descrivono rappresentazioni interne, come la memorizzazione della memoria in rappresentazione binaria.

Domande (MC 3.4.B.5)

1. Qual è il ruolo fondamentale dei tipi di dati in un linguaggio di programmazione e perché sono considerati elementi costitutivi della programmazione?
2. Fornire un elenco completo dei principali tipi di dati utilizzati nella programmazione e delinearne le caratteristiche.
3. Scegliete uno scenario di programmazione specifico e spiegate quali tipi di dati sarebbero più adatti, giustificando le vostre scelte in base alle loro caratteristiche.
4. Distinguere tra rappresentazioni interne ed esterne dei tipi di dati, utilizzando esempi per illustrare la propria comprensione.
5. Nel contesto dello sviluppo di un'applicazione web che gestisce i dati degli utenti, proporre un tipo di dati appropriato per la memorizzazione dell'età degli utenti, garantendo efficienza e precisione.
6. Elaborare il ruolo dei tipi di dati avanzati come i puntatori e le strutture/record nella programmazione.
7. Fornite un esempio di scenario in cui l'uso di un array è più vantaggioso rispetto ad altri tipi di dati e giustificate la vostra scelta.

Variabili e operatori (MC 3.4.B.6)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Variabili e operatori. Codice: MC 3.2.B.6
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 4 - Massimo 6 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	INTERMEDIO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.B.6)

Risultati di apprendimento (rif. LO di livello 3-4 3.4.23 - 3.2.24):

- È consapevole di come dichiarare una variabile.
- È consapevole di come utilizzare gli operatori.

Descrizione (MC 3.4.B.6)

Al livello intermedio, gli sviluppatori dovrebbero comprendere che la dichiarazione delle variabili è un passo fondamentale nella programmazione, assicurandosi che siano definite prima dell'uso. Riconoscono un identificatore come meccanismo di denominazione di varie entità di programmazione, rispettando le regole definite, come ad esempio i set di caratteri ammessi (lettere, cifre, trattini bassi) e le considerazioni sulla lunghezza. Inoltre, dimostrano di conoscere le convenzioni di denominazione delle variabili, essenziali per mantenere la chiarezza e la coerenza del codice nei vari linguaggi di programmazione.

I programmatori devono sapere che gli operatori sono rappresentazioni simboliche o parole chiave che facilitano operazioni specifiche sugli operandi, che possono essere valori o variabili. Devono conoscere le categorie comuni di operatori, tra cui aritmetici, di confronto, logici, di assegnazione, bitwise, condizionali, cast, instanceof, member access e address-of. Conoscono i tipi di operatori, compresi quelli unari, binari e ternari, riconoscendo i loro ruoli distinti nella logica di programmazione. Inoltre, comprendono il concetto di notazione degli operatori e la loro variabilità (infixo, prefixo, postfixo), riconoscendo che lo stile di notazione dipende dal linguaggio.

Domande (MC 3.4.B.6)

1. Spiegare il significato della dichiarazione delle variabili nella programmazione.
2. Illustrate le regole per la definizione di un nome di variabile e fornite un esempio di nome di variabile valido, tenendo conto dei caratteri consentiti.
3. Definire cosa sia un identificatore nel contesto della programmazione.
4. Descrivere il ruolo degli operatori in un linguaggio di programmazione.
5. Enumerare le categorie comuni di operatori e fornire esempi per ogni categoria, illustrando le loro distinte funzionalità.
6. Distinguere tra operatori unari, binari e ternari.
7. Spiegare il concetto di notazione degli operatori e le sue varianti (infixo, prefixo, postfixo).

Dichiarazioni sul flusso di controllo (MC 3.4.B.7)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Dichiarazioni sul flusso di controllo. Codice: MC 3.2.B.7
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 4 - Massimo 6 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	INTERMEDIO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.B.7)

Risultati di apprendimento (rif. Livello 3-4 LOs 3.4.27):

- Elencate e spiegate le istruzioni del flusso di controllo.

Descrizione (MC 3.4.B.7)

Al livello intermedio di competenza nella programmazione, le persone devono capire che le istruzioni di flusso di controllo sono fondamentali per progettare programmi che eseguono azioni diverse in base a condizioni specifiche. Un programmatore di livello intermedio possiede le conoscenze necessarie per utilizzare le varie istruzioni di flusso di controllo che modellano la sequenza di esecuzione di un programma. Ciò include la comprensione del significato delle istruzioni di flusso di controllo, che sono costrutti integrali che dettano l'ordine di esecuzione del programma. Queste istruzioni forniscono i mezzi per implementare la ramificazione condizionale e il comportamento di looping. Gli sviluppatori devono conoscere i vari tipi di istruzioni sul flusso di controllo, comprese le istruzioni condizionali (if, if... else), le istruzioni di loop (do... while, while, for), le istruzioni di controllo del loop (break, continue), le istruzioni di controllo (goto) e le istruzioni di gestione delle eccezioni (try... catch).

Devono essere in grado di comprendere e applicare la logica che sta alla base di ogni istruzione del flusso di controllo. Ciò implica la conoscenza delle condizioni in cui si verifica la ramificazione, dei criteri di iterazione dei cicli e del ruolo delle istruzioni di controllo nell'alterare il flusso del programma.

I programmatori intermedi devono dimostrare di essere in grado di prendere decisioni informate su quando utilizzare specifiche istruzioni di flusso di controllo. Comprendono le implicazioni delle loro scelte sul comportamento del programma, sulla leggibilità e sulla manutenzione.

Domande (MC 3.4.B.7)

1. Spiegare il significato delle dichiarazioni di flusso di controllo nella programmazione e come contribuiscono alla struttura complessiva e alla logica di un programma.
2. Fornite un esempio di scenario in cui l'uso efficace del flusso di controllo migliora la funzionalità del programma.
3. Elencate e spiegate brevemente ogni tipo di istruzione del flusso di controllo, comprese quelle condizionali, di loop, di controllo, di controllo e di gestione delle eccezioni.
4. Illustrare le situazioni in cui ciascun tipo è utilizzato in modo appropriato.
5. Fornite un'analogia o un esempio del mondo reale per illustrare l'uso di una dichiarazione di flusso di controllo.
6. Per quanto riguarda i cicli (do...while, while, for), descrivere le differenze fondamentali e gli scenari in cui ciascuno di essi è più adatto.
7. Spiegare lo scopo e il funzionamento delle istruzioni di gestione delle eccezioni (try...catch) per controllare il flusso del programma durante le situazioni di errore.

Eeguire il file sorgente (MC 3.4.B.8)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Eeguire il file sorgente. Codice: MC 3.2.B.8
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 2 - Massimo 4 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	INTERMEDIO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.B.8)

Risultati di apprendimento (rif. LO di livello 3-4 3.4.28 - 3.2.30):

- È consapevole di come scrivere un semplice file sorgente.
- È consapevole di come eseguire un file sorgente.

Descrizione (MC 3.4.B.8)

Al livello intermedio di competenza, gli sviluppatori dovrebbero essere consapevoli della creazione di un nuovo file sorgente con l'estensione corretta per il linguaggio di programmazione scelto. Sono in grado di applicare le regole della sintassi per creare una semplice applicazione che può includere variabili, funzioni e dichiarazioni sul flusso di controllo. Conoscendo bene gli editor di testo o gli ambienti di sviluppo integrati (IDE), seguono un processo sistematico: creano un nuovo file, scrivono il codice, lo salvano con un nome significativo e l'estensione appropriata e selezionano una posizione per un facile recupero. Lo sviluppatore è in grado di organizzare il codice in progetti e di gestire i file di progetto. Questo include la creazione di strutture di progetto, l'aggiunta di dipendenze e la configurazione di percorsi di compilazione.

Le competenze si estendono alla comprensione dei passaggi sfumati necessari in base al linguaggio di programmazione, agli strumenti e all'ambiente di sviluppo in uso. Sono in grado di compilare/interpretare, se necessario, per generare un file eseguibile. Durante l'esecuzione del programma, le persone sono in grado di navigare sia nelle interfacce a riga di comando che negli IDE. Nell'interfaccia a riga di comando, utilizzano il comando appropriato, mentre nell'interfaccia dell'IDE sfruttano l'opzione adatta per eseguire l'applicazione.

Domande (MC 3.4.B.8)

1. Potete descrivere le fasi essenziali della creazione di un nuovo file sorgente per un linguaggio di programmazione?
2. Come si decide un nome significativo per il file di origine e quali sono le considerazioni da fare per scegliere l'estensione corretta del file?
3. Come si fa a decidere una posizione adatta sul computer per salvare il file e recuperarlo facilmente in seguito?
4. Potete illustrare i passaggi generali per eseguire un file sorgente, tenendo conto del linguaggio di programmazione, degli strumenti installati e dell'ambiente di sviluppo che state utilizzando?
5. Spiegate il concetto di compilazione nella programmazione e in quali circostanze dovrete compilare il vostro codice sorgente prima di eseguirlo?
6. Descrivere il processo di esecuzione di un programma in un'interfaccia a riga di comando (terminale)/ambiente di sviluppo integrato (IDE).
7. Descrivete i passi da compiere per organizzare il codice in progetti, compresa la creazione di una struttura di progetto, l'aggiunta di dipendenze e la configurazione dei percorsi di compilazione.

LIVELLO AVANZATO
(Livello 5 e Livello 6)



Metodi/funzioni (MC 3.4.C.1)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Metodi/funzioni. Codice: MC 3.2.C.1
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 4 - Massimo 6 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	AVANZATO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.C.1)

Risultati di apprendimento (rif. Livello 5-6 LO 3.4.31 - 3.2.33):

- Dichiarare metodi/funzioni.
- Definire metodi/funzioni.
- Applicare metodi/funzioni.

Descrizione (MC 3.4.C.1)

Al livello avanzato di competenza nella programmazione, gli sviluppatori devono capire che la dichiarazione dei metodi comporta il riconoscimento di elementi chiave come il nome del metodo, il tipo di ritorno, i parametri e il corpo del metodo. Questa conoscenza si estende a diversi linguaggi di programmazione, garantendo la coerenza delle pratiche di dichiarazione dei metodi.

La competenza si estende alla definizione dei metodi, dove l'esperienza consiste nel creare il comportamento, la logica e la funzionalità del metodo in base alla dichiarazione precedente. Ciò comprende la creazione del corpo del metodo, la formulazione degli algoritmi e della logica per l'esecuzione del task e, se del caso, l'inclusione di un'istruzione di ritorno per specificare il valore inviato al chiamante.

La competenza nell'applicazione si dimostra attraverso l'abilità di invocare metodi in diverse parti dell'applicazione, eseguendo azioni designate. Ciò comporta una comprensione completa di come i metodi interagiscono all'interno della base di codice, contribuendo alla funzionalità e all'efficienza complessiva del programma.

Domande (MC 3.4.C.1)

1. Quali sono gli elementi essenziali di una dichiarazione di metodo nella programmazione?
2. Spiegare i componenti della definizione di un metodo nella programmazione, compreso il significato del corpo del metodo, dell'algoritmo e dell'istruzione di ritorno.
3. Fornite un esempio di scenario in cui la chiamata di un metodo è fondamentale per l'esecuzione di azioni specifiche all'interno di un programma. Descrivete i passaggi necessari per richiamare un metodo da diverse parti della base di codice.
4. In diversi linguaggi di programmazione, quali sono gli elementi chiave comuni che si trovano costantemente nelle dichiarazioni dei metodi?
5. Dimostrare di aver compreso l'interazione tra dichiarazione e definizione dei metodi.

Operazioni di ingresso/uscita (I/O) (MC 3.4.C.2)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Operazioni di ingresso/uscita (I/O). Codice: MC 3.2.C.2
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 4 - Massimo 6 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	AVANZATO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.C.2)

Risultati di apprendimento (rif. Livello 5-6 LO 3.4.34 - 3.2.35):

- È consapevole delle operazioni di input/output (I/O).
- Eseguire operazioni di ingresso/uscita (I/O).

Descrizione (MC 3.4.C.2)

Al livello avanzato di competenza nella programmazione, gli sviluppatori devono sapere che le operazioni di input/output (I/O) nella programmazione sono i mezzi per leggere dati da fonti esterne o inviare dati a destinazioni esterne, come interagire con utenti, file, risorse di rete, database e altri sistemi. Riconosce il ruolo fondamentale delle operazioni di I/O nell'elaborazione dei dati. Le competenze avanzate nelle operazioni di I/O dimostrano la capacità di eseguire varie operazioni di I/O, tra cui la lettura di input da diverse fonti (tastiera, file e connessioni di rete), la scrittura di output (su console, file, rete e database), l'implementazione di output formattati per visualizzazioni a lettura umana, l'applicazione di tecniche di gestione degli errori per gestire le eccezioni durante le operazioni di I/O, la conduzione di processi di serializzazione e deserializzazione per l'archiviazione o la trasmissione di dati, la comprensione e l'utilizzo di flussi standard (input - stdin, output - stdout ed error - stderr).

Domande (MC 3.4.C.2)

1. Qual è lo scopo delle operazioni di input/output (I/O) nella programmazione e perché sono fondamentali per le applicazioni software che interagiscono con sistemi esterni?
2. Dimostrare le fasi di esecuzione delle operazioni di I/O. Fornire esempi di lettura di input da diverse fonti (tastiera, file, rete) e di scrittura di output verso diverse destinazioni (console, file, rete, database).
3. Come si possono visualizzare i dati in un formato leggibile, applicando la formattazione a numeri e date durante le operazioni di output?
4. Definire la serializzazione e la deserializzazione nel contesto della programmazione.
5. Come vengono utilizzati i flussi di input (stdin), output (stdout) ed errore (stderr) nella programmazione?

Librerie e moduli (MC 3.4.C.3)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Librerie e moduli. Codice: MC 3.2.C.3
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	AVANZATO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.C.3)

Risultati di apprendimento (rif. Livello 5-6 LO 3.4.36 - 3.2.37):

- È consapevole delle librerie e dei moduli.
- Operare con librerie e moduli.

Descrizione (MC 3.4.C.3)

Al livello avanzato di competenza nella programmazione, gli sviluppatori devono essere in grado di definire il concetto di librerie e moduli nella programmazione. Spiegare il ruolo delle librerie come raccolte di codice precompilato con funzionalità specifiche e dei moduli come unità di codice autonome. Dovrebbero inoltre essere in grado di discutere l'importanza di fattori quali la compatibilità, le licenze, la documentazione e il supporto della comunità quando si considera l'uso di librerie e moduli.

In secondo luogo, devono spiegare l'impatto dell'utilizzo di librerie e moduli sulla velocità e sull'efficienza dello sviluppo del software. Discutere come l'utilizzo di codice prescritto permetta agli sviluppatori di concentrarsi sulla logica di base e di implementare più rapidamente gli aspetti unici di un'applicazione.

Domande (MC 3.4.C.3)

1. Spiegate il concetto di librerie e moduli nella programmazione. Cosa distingue una libreria da un modulo e come vengono utilizzati per migliorare le applicazioni software?
2. Discutere l'importanza di considerare fattori come la compatibilità, le licenze, la documentazione e il supporto della comunità quando si lavora con librerie e moduli.
3. Descrivere i vantaggi del riutilizzo del codice attraverso librerie e moduli.
4. In che modo l'utilizzo di librerie e moduli può contribuire a velocizzare lo sviluppo?
5. Come può un programmatore partecipare attivamente e trarre vantaggio dagli sforzi della comunità per sviluppare e migliorare le biblioteche?

Paradigmi di programmazione (MC 3.4.C.4)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Paradigmi di programmazione. Codice: MC 3.2.C.4
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	AVANZATO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 50%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.C.4)

Risultati di apprendimento (rif. Livello 5-6 LOs 3.4.38):

- Enumerare e spiegare i paradigmi di programmazione.

Descrizione (MC 3.4.C.4)

Al livello avanzato di competenza nella programmazione, gli sviluppatori devono sapere che i paradigmi di programmazione sono approcci fondamentali alla strutturazione e all'organizzazione del codice, che rappresentano diverse filosofie e metodologie per progettare e implementare il software. Ogni paradigma fornisce un modo distinto di pensare alla progettazione dei programmi, enfatizzando principi e pratiche particolari.

I programmatori devono elencare e spiegare le caratteristiche dei principali paradigmi di programmazione, tra cui la programmazione imperativa, orientata agli oggetti, funzionale, logica, concorrente, parallela, procedurale, guidata dagli eventi e strutturata. Evidenziano i concetti chiave e gli esempi associati a ciascun paradigma. Inoltre, devono essere in grado di evidenziare il fatto che molti linguaggi di programmazione e framework moderni supportano una combinazione di paradigmi di programmazione.

Domande (MC 3.4.C.4)

1. Enumerare i principali paradigmi di programmazione e spiegare brevemente le caratteristiche di ciascuno. Fornite un esempio di linguaggio per ogni paradigma.
2. Descrivere le caratteristiche principali della programmazione imperativa. Fornire esempi di linguaggi di programmazione che seguono questo paradigma.
3. Spiegare i principi fondamentali della programmazione funzionale. Fornire esempi di linguaggi di programmazione funzionale.
4. Discutere le caratteristiche della programmazione orientata agli oggetti (OOP). Fornire esempi di linguaggi OOP.
5. Esempificare i moderni linguaggi di programmazione o i framework che supportano una combinazione di paradigmi di programmazione.
6. Fornite un esempio di scenario in cui una combinazione di paradigmi potrebbe essere vantaggiosa.

Programmazione imperativa e orientata agli oggetti (MC 3.4.C.5)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Programmazione imperativa e orientata agli oggetti. Codice: MC 3.2.C.5
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	AVANZATO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.C.5)

Risultati di apprendimento (rif. Livello 5-6 LO 3.4.39 - 3.2.40):

- Spiegare i concetti di programmazione procedurale imperativa.
- Spiegare i concetti di programmazione orientata agli oggetti (OOP).

Descrizione (MC 3.4.C.5)

Al livello avanzato di competenza nella programmazione, gli sviluppatori devono sapere che la programmazione procedurale imperativa, un paradigma fondamentale, si concentra sulla sequenza di passi e procedure per portare a termine i compiti. I concetti chiave comprendono sequenze, variabili, strutture di controllo, programmazione strutturata, modularità e procedure/funzioni. Una solida comprensione di questi concetti consente ai programmatori avanzati di creare codice ben organizzato, modulare ed estensibile che affronta in modo efficiente problemi complessi.

Inoltre, i programmatori devono sapere che la programmazione orientata agli oggetti (OOP) è un paradigma che enfatizza gli oggetti, le classi e i metodi per l'organizzazione del codice e la risoluzione dei problemi. I concetti chiave comprendono gli oggetti e le classi, l'incapsulamento, l'ereditarietà, il polimorfismo, l'overriding dei metodi, l'overloading dei metodi, i costruttori e i distruttori, il passaggio di messaggi e i modificatori di visibilità.

Domande (MC 3.4.C.5)

1. Spiegare il concetto di "sequenziamento" nella programmazione procedurale imperativa.
2. In che modo la "modularità" contribuisce all'efficacia del codice procedurale imperativo?
3. Discutere l'importanza della "programmazione strutturata" per migliorare la manutenibilità e la leggibilità del codice.
4. Elaborare il concetto di "incapsulamento" nell'OOP e fornire un esempio che illustri come migliora la sicurezza e l'organizzazione del codice.
5. Discutete i principi di "ereditarietà" nell'OOP e spiegate come facilita il riutilizzo del codice e promuove una struttura gerarchica.
6. Descrivere il ruolo dei "costruttori e distruttori" nell'OOP, sottolineando quando e perché sono essenziali nella progettazione delle classi.

Programmazione logica e funzionale (MC 3.4.C.6)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Logica e programmazione funzionale. Codice: MC 3.2.C.6
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	AVANZATO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.C.6)

Risultati di apprendimento (rif. Livello 5-6 LO 3.4.41 - 3.2.42):

- Spiegare i concetti di logica dichiarativa.
- Spiegare i concetti di funzionale dichiarativo.

Descrizione (MC 3.4.C.6)

Al livello avanzato di competenza nella programmazione di applicazioni, gli sviluppatori dovrebbero possedere una profonda comprensione della programmazione logica dichiarativa, che si concentra sull'espressione di relazioni e fatti senza specificare esplicitamente il flusso di controllo. Questo paradigma, spesso esemplificato da linguaggi come il Prolog, consente agli sviluppatori di dichiarare la logica e lasciare che sia il sistema a determinare la sequenza di esecuzione. Il programmatore è abile nel costruire sistemi basati sulla logica, nell'utilizzare i fatti e nel definire le relazioni senza la necessità di istruzioni procedurali esplicite. Questa conoscenza è fondamentale per i compiti che coinvolgono i sistemi basati su regole, dove l'espressione della logica delle relazioni ha la precedenza sulla specificazione dell'esecuzione passo-passo.

La programmazione funzionale dichiarativa, per il programmatore avanzato, è incentrata sull'espressione della computazione come funzioni matematiche senza fare affidamento su stati mutabili o effetti collaterali. Il paradigma funzionale, esemplificato da linguaggi come Haskell e Lisp, promuove una rappresentazione più concisa e astratta degli algoritmi. Il programmatore è in grado di creare sistemi robusti, modulari e scalabili sfruttando la potenza delle funzioni di ordine superiore, l'immutabilità e l'attenzione alle computazioni pure.

Domande (MC 3.4.C.6)

1. Può articolare il principio chiave della programmazione logica dichiarativa e come si differenzia dai paradigmi imperativi, che enfatizzano l'espressione di relazioni e fatti?
2. Fornite un esempio di scenario in cui la programmazione logica dichiarativa, come Prolog, sarebbe più vantaggiosa di un approccio imperativo.
3. Descrivere i principi fondamentali della programmazione funzionale dichiarativa.
4. In che modo il paradigma della programmazione funzionale migliora la chiarezza e la manutenibilità del codice rispetto agli approcci imperativi o orientati agli oggetti?

Programmazione concorrente (MC 3.4.C.7)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Programmazione concorrente. Codice: MC 3.2.C.7
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 3 - Massimo 5 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	AVANZATO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.C.7)

Risultati di apprendimento (rif. Livello 5-6 LO 3.4.43 - 3.2.44):

- Elencare le tecniche di concomitanza.
- Eseguire la concomitanza.

Descrizione (MC 3.4.C.7)

Una competenza avanzata nella programmazione di applicazioni implica una comprensione completa delle tecniche di concorrenza e la capacità di implementarle in modo efficace. Gli sviluppatori devono comprendere le tecniche di multithreading (i thread operano all'interno di un singolo processo, condividendo lo spazio di memoria) e di multi-processing (uso di processi separati con spazi di memoria distinti per la concomitanza). Devono identificare i compiti indipendenti e valutare quali parti del programma possono essere eseguite simultaneamente senza conflitti. Quindi devono determinare le risorse condivise e applicare meccanismi di sincronizzazione (serrature, semafori) per gestire l'accesso e prevenire le corse dei dati e garantire un accesso ai dati sicuro per i thread, utilizzando misure di sicurezza e di sincronizzazione.

Domande (MC 3.4.C.7)

1. Descrivere il concetto fondamentale di concorrenza e il suo ruolo nell'efficienza dei programmi.
2. Distinguere tra multithreading e multi-processing, spiegando come ciascuna tecnica realizza la concorrenza.
3. Scegliete tra multithreading e multi-processing in base ai requisiti e ai vincoli di una determinata applicazione.
4. Dato un frammento di codice, identificare potenziali compiti indipendenti adatti all'esecuzione concorrente.
5. Valutare un pezzo di codice e identificare le risorse condivise che possono portare a gare di dati. Proporre meccanismi di sincronizzazione per mitigare questi problemi.
6. Analizzare un frammento di codice e identificare le aree potenziali in cui possono verificarsi deadlock o condizioni di gara. Proporre soluzioni per prevenire o risolvere questi problemi.

Test e debug (MC 3.4.C.8)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Test e debug. Codice: MC 3.2.C.8
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 8 - Massimo 10 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	AVANZATO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.C.8)

Risultati di apprendimento (rif. Livello 5-6 LO 3.4.45 - 3.2.48):

- Test e debug.
- Eseguire il debug di un'applicazione.
- Testare un'applicazione.

Descrizione (MC 3.4.C.8)

I programmatori di livello avanzato possiedono una comprensione e un'applicazione completa delle fasi di test e debug. Riconoscono il testing come un processo sistematico per identificare i difetti, assicurando che il software aderisca alle specifiche e si comporti come previsto. Inoltre, intendono il debugging come la risoluzione mirata dei difetti, che comprende l'esame del codice, il tracciamento dell'esecuzione e l'identificazione della causa principale del problema. Conoscere e applicare tecniche di debugging come l'uso di istruzioni di stampa per produrre valori di variabili, il debugging interattivo in IDE o strumenti, l'utilizzo di punti di interruzione e l'ispezione delle variabili, l'implementazione di log con librerie per registrare i dettagli dell'esecuzione del programma per un'analisi approfondita, la collaborazione per la revisione del codice.

In secondo luogo, a livello avanzato, gli sviluppatori dimostrano di possedere competenze in metodologie di test come i test unitari (automatizzati tramite framework), i test di integrazione (per convalidare le interazioni tra unità e moduli diversi), i test funzionali (scenari), i test di regressione (per prevenire l'introduzione di difetti con le nuove modifiche), i test di prestazione (per valutare la reattività del software in condizioni diverse), i test di sicurezza (per identificare le vulnerabilità e mitigare i rischi potenziali).

Domande (MC 3.4.C.8)

1. Come definirebbe il ruolo dei test e del debug nel ciclo di vita dello sviluppo del software?
2. Descrivete le tecniche di debug che impieghereste per identificare e risolvere i problemi in una base di codice complessa.
3. Quando ci si trova di fronte a un bug in un'applicazione, come si fa a sfruttare le istruzioni di stampa, il debug interattivo negli IDE, le librerie di log e le revisioni del codice per identificare e risolvere in modo efficiente la causa principale del problema?
4. Spiegate l'importanza dei vari tipi di test (unit testing, integration testing, functional testing, regression testing, performance testing e security testing) per garantire un'applicazione software robusta e affidabile. In che modo questi tipi si completano a vicenda?
5. Come affrontate i processi di test e debug, considerando la loro natura iterativa?

Linguaggi di programmazione compilati e interpretati (MC 3.4.C.9)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Linguaggi di programmazione compilati e interpretati. Codice: MC 3.2.C.9
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 6 - Massimo 8 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	AVANZATO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.C.9)

Risultati di apprendimento (rif. Livello 5-6 LO 3.4.49 - 3.2.52):

- Spiegare le differenze tra compilato e interpretato (scripting).
- Caratteristiche del linguaggio compilato.
- Caratteristiche del linguaggio interpretato.
- È consapevole del tipo di linguaggio di programmazione.

Descrizione (MC 3.4.C.9)

Una competenza avanzata nelle applicazioni di programmazione implica la distinzione tra linguaggi compilati e interpretati. I linguaggi compilati sono sottoposti a una fase di compilazione precedente all'esecuzione, che traduce il codice sorgente in un efficiente codice macchina o bytecode. I linguaggi interpretati, invece, eseguono il codice in tempo reale, riga per riga, offrendo cicli di sviluppo rapidi e facilità di debug.

I professionisti avanzati riconoscono le caratteristiche dei linguaggi compilati, sottolineando le prestazioni superiori, la possibilità di esecuzioni multiple senza ricompilazione, ma la ridotta portabilità dovuta alla compilazione specifica della piattaforma. Ne sono un esempio C, C++, Rust e Go. I linguaggi interpretati danno la priorità alla facilità di sviluppo. Presentano prestazioni generalmente più lente, ma i moderni interpreti incorporano ottimizzazioni. In particolare, vantano una maggiore portabilità, poiché lo stesso codice può essere eseguito su diverse piattaforme. Ne sono un esempio Python, Ruby, JavaScript e PHP.

Inoltre, i professionisti avanzati comprendono la distinzione tra linguaggi tipizzati staticamente e dinamicamente. I linguaggi a tipizzazione statica impongono le dichiarazioni dei tipi a tempo di compilazione, garantendo la correttezza dei tipi, come nel caso di Java o C++. I linguaggi a tipizzazione dinamica, come Python o JavaScript, determinano i tipi in fase di esecuzione, offrendo la possibilità di cambiare dinamicamente i tipi delle variabili durante l'esecuzione.

Domande (MC 3.4.C.9)

1. Descrivere le principali differenze tra linguaggi compilati e interpretati nel contesto della fase di compilazione pre-esecuzione.
2. Considerando le caratteristiche dei linguaggi compilati, discutete il compromesso tra prestazioni superiori e portabilità ridotta.
3. Spiegare i vantaggi dei linguaggi interpretati, sottolineando la loro esecuzione in tempo reale e i rapidi cicli di sviluppo.
4. Discutere le implicazioni dei linguaggi tipizzati staticamente, sottolineando il ruolo del controllo di tipo a tempo di compilazione.
5. In che modo la tipizzazione dinamica consente di modificare i tipi di variabili durante l'esecuzione del programma e quali vantaggi offre questa flessibilità a determinati scenari di sviluppo?

Linguaggi di markup (MC 3.4.C.10)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Linguaggi di markup. Codice: MC 3.2.C.10
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 6 - Massimo 8 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	AVANZATO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.C.10)

Risultati di apprendimento (rif. Livello 5-6 LO 3.4.53 - 3.2.54):

- Spiegare i concetti dei linguaggi di markup.
- Trasformazione e styling dei linguaggi di markup.

Descrizione (MC 3.4.C.10)

Una competenza avanzata nelle applicazioni di programmazione implica una comprensione approfondita dei linguaggi di markup. Questi linguaggi, come HTML, XML, Markdown e LaTeX, servono ad annotare il testo, definendone la struttura, la presentazione e il comportamento all'interno di un documento. La padronanza dei linguaggi di markup comprende la comprensione dei concetti essenziali, tra cui i tag di markup, gli attributi, la nidificazione, la validazione e la garanzia di interoperabilità tra piattaforme e dispositivi diversi.

Inoltre, un professionista avanzato dimostra di saper trasformare e stilizzare i contenuti utilizzando tecnologie come i fogli di stile a cascata (CSS) per i documenti web e le trasformazioni del linguaggio Extensible Stylesheet (XSLT) per i documenti XML. Questa abilità consiste nello sfruttare le tecnologie di trasformazione per migliorare la presentazione visiva e la rappresentazione strutturale dei contenuti. L'applicazione competente di queste tecniche garantisce la creazione di contenuti digitali accessibili, ben organizzati ed esteticamente gradevoli in vari contesti e piattaforme.

Domande (MC 3.4.C.10)

1. È in grado di articolare i concetti fondamentali dei linguaggi di markup?
2. Dimostra di comprendere il ruolo dei linguaggi di markup nel definire la struttura, la presentazione e il comportamento del testo all'interno di un documento.
3. È in grado di spiegare come i linguaggi di markup contribuiscano all'interoperabilità tra piattaforme e dispositivi diversi.
4. Dimostra di saper utilizzare i fogli di stile a cascata (CSS) per lo styling dei documenti web.
5. Dimostra competenza nell'utilizzo di Extensible Stylesheet Language Transformations (XSLT) per la trasformazione e lo styling di documenti XML.
6. Mostra la capacità di applicare efficacemente le tecnologie di trasformazione per migliorare la presentazione visiva e la rappresentazione strutturale dei contenuti.

LIVELLO ESPERTO
(Livello 7 e Livello 8)



Progettare la soluzione di un problema complesso (MC 3.4.D.1)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Progettare soluzioni per problemi complessi. Codice: MC 3.2.D.1
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 8 - Massimo 10 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	ESPERTO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.D.1)

Risultati di apprendimento (rif. Livello 7-8 LO 3.4.5% - 3.2.57):

- Stabilire dichiarazioni e requisiti per problemi complessi con definizione limitata.
- Progettare soluzioni per risolvere problemi complessi con molti fattori interagenti.
- Progetta l'implementazione di sistemi informatici complessi.

Descrizione (MC 3.4.D.1)

I professionisti altamente specializzati sono in grado di formulare domande critiche per svelare le complessità di problemi poco definiti o in evoluzione in domini specializzati. Dimostrano la capacità di affinare iterativamente i problemi collaborando con esperti del settore e analizzando le esigenze degli utenti. Integra approcci sistematici e strategici per progettare soluzioni a problemi complessi con numerosi fattori interagenti. Dimostra competenza nella scomposizione dei problemi, nell'analisi dei dati, nella modellazione matematica e nella collaborazione tra discipline. Applica tecniche di apprendimento automatico e di intelligenza artificiale per analizzare e prevedere interazioni complesse.

Progetta e integra componenti hardware, software e di rete per creare sistemi informatici funzionali e sicuri. Dimostra responsabilità nel soddisfare i requisiti di dimensione, prestazioni e complessità, garantendo scalabilità e facilità di manutenzione. Valuta e raccomanda nuove tecnologie, tenendo conto di fattori quali le caratteristiche di sicurezza, il supporto a lungo termine, l'integrazione e i vincoli di budget. Prende decisioni informate sulla selezione di piattaforme, linguaggi e framework di destinazione in base a considerazioni sull'ecosistema e ai requisiti del progetto.

Gestisce le risorse personali in modo efficace, considerando la disponibilità di librerie, le caratteristiche di sicurezza, il supporto a lungo termine, l'integrazione e i vincoli di bilancio. Mantiene una comunicazione aperta e trasparente con le parti interessate, garantendo pratiche di sviluppo responsabili.

Domande (MC 3.4.D.1)

1. Come affrontate il perfezionamento iterativo delle dichiarazioni del problema, soprattutto quando collaborate con esperti del dominio e considerate le esigenze degli utenti?
2. Descrivete una situazione in cui avete applicato con successo un approccio sistematico e strategico per progettare soluzioni a un problema complesso con molteplici fattori interagenti.
3. Come garantire la competenza nella scomposizione del problema, nell'analisi dei dati, nella modellazione matematica e nella collaborazione tra diverse discipline nel processo di progettazione della soluzione?
4. Fornite esempi di come avete applicato tecniche di apprendimento automatico e di intelligenza artificiale per analizzare e prevedere interazioni complesse nei vostri progetti.
5. Come si fa a garantire che il sistema informatico progettato soddisfi i requisiti di dimensione, prestazioni e complessità, assicurando scalabilità e facilità di manutenzione?
6. Come si fa a prendere decisioni informate quando si selezionano piattaforme, linguaggi e framework di destinazione, tenendo conto delle considerazioni sull'ecosistema e dei requisiti del progetto?

Project management (MC 3.4.D.2)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Project management. Codice: MC 3.2.D.2
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 8 - Massimo 10 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	ESPERTO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.D.2)

Risultati di apprendimento (rif. Livello 7-8 LO 3.4.58 - 3.2.60):

- Supervisiona l'implementazione di sistemi informatici complessi.
- Supervisionare la conformità alle specifiche.
- Implementare i metodi di gestione dei progetti.

Descrizione (MC 3.4.D.2)

L'esperienza altamente specializzata nella programmazione di applicazioni implica la supervisione dell'implementazione di sistemi informatici complessi, con una conoscenza approfondita di diversi linguaggi e tecnologie. Il manager pone l'accento sul rispetto delle specifiche, assicurando una precisa aderenza alle best practice di codifica, alle strutture di progetto, al controllo delle versioni e ad altri concetti fondamentali per la creazione di applicazioni sofisticate e mantenibili.

L'implementazione dei metodi di gestione dei progetti comprende compiti quali la pianificazione completa del progetto, la definizione degli obiettivi e lo sviluppo delle scadenze. La gestione del team comporta l'assemblaggio di squadre qualificate, l'assegnazione dei ruoli e la promozione della collaborazione. Il project manager si impegna nella gestione del rischio, nell'assicurazione della qualità e nel monitoraggio e controllo continui, assicurando che il progetto rimanga in linea. La gestione delle risorse e la documentazione meticolosa contribuiscono al successo dell'esecuzione del progetto software, dimostrando un approccio olistico al ciclo di vita dello sviluppo.

Domande (MC 3.4.D.2)

1. Come avete garantito la conformità alle specifiche, alle best practice di codifica e alle strutture di progetto nei vostri progetti precedenti.
2. In che modo questo impegno ha contribuito alla creazione di applicazioni sofisticate e mantenibili?
3. Spiegate il vostro approccio alla pianificazione completa del progetto, compreso il modo in cui definite gli obiettivi del progetto, sviluppate le scadenze e allocate le risorse.
4. In che modo la vostra competenza nella gestione dei progetti ha contribuito al successo dell'esecuzione di un progetto software?
5. Descrivete la vostra esperienza nell'assemblaggio di team qualificati, nell'assegnazione dei ruoli e nella promozione della collaborazione in un progetto di sviluppo software.
6. Come gestite la gestione del rischio e la garanzia di qualità nei progetti di sviluppo software?

Leadership. Creatività (MC 3.4.D.3)

Informazioni di base

Identificazione dell'allievo	Qualsiasi cittadino
Titolo e codice della microcredenziale	Leadership. Creatività. Codice: MC 3.2.D.3
Paese(i)/Regione(i) dell'emittente	IRLANDA, ITALIA, CIPRO, GRECIA, ROMANIA http://dsw.projectsgallery.eu
Ente/i di assegnazione	Consorzio DSW Numero del progetto: 101087628
Data di emissione	Novembre 2023
Carico di lavoro figurativo necessario per raggiungere i risultati dell'apprendimento	Minimo 8 - Massimo 10 ore
Livello dell'esperienza di apprendimento che porta al microcredenziale	ESPERTO
Tipo di valutazione	Domande contrassegnate automaticamente. Numero di domande: 10 Punteggio di superamento: 75%
Forma di partecipazione all'attività di apprendimento	Online Asincrono
Tipo di garanzia della qualità utilizzata per sostenere la microcredenziale	Revisione tra pari

Risultati di apprendimento (MC 3.4.D.3)

Risultati di apprendimento (rif. Livello 7-8 LO 3.4.61 - 3.2.62):

- Guidare altri nell'analisi e nello sviluppo di applicazioni.
- Proporre nuove idee e processi al settore.

Descrizione (MC 3.4.D.3)

Al livello di competenza altamente specializzata nella programmazione di applicazioni, il professionista assume un ruolo di leadership guidando altri nell'analisi e nello sviluppo di applicazioni. Contribuisce attivamente alle pratiche professionali, favorendo l'apprendimento continuo e l'aggiornamento delle conoscenze all'interno del team. Attraverso il tutoraggio, i progetti di collaborazione e la creazione di documentazione e best practice, consente alla propria organizzazione di eccellere. Ciò comporta lo sviluppo di pattern riutilizzabili, la partecipazione a progetti open-source e la promozione di pratiche etiche e conformi, dell'automazione e dell'integrazione continua.

Inoltre, l'esperto incoraggia l'innovazione proponendo nuove idee e processi al settore. Sottolinea l'importanza della creatività, della risoluzione dei problemi, della ricerca e della collaborazione interdisciplinare. La loro attenzione si estende al miglioramento continuo, all'efficienza, alla riduzione dei costi, alla scalabilità, alla progettazione incentrata sull'utente, alla sostenibilità e alla gestione del rischio. Nel complesso, questo professionista si sforza di apportare cambiamenti positivi, progressi ed efficienze, assicurando che la sua organizzazione rimanga competitiva e adattabile in un ambiente in rapida evoluzione.

Domande (MC 3.4.D.3)

1. In che modo il professionista contribuisce attivamente all'apprendimento continuo e all'aggiornamento delle conoscenze all'interno del team e quali strategie utilizza per promuovere una cultura dell'apprendimento e dello sviluppo?
2. Può fornire esempi di progetti di collaborazione e dell'impatto del suo contributo sul successo del team e sui risultati del progetto?
3. In che modo l'individuo consente alla propria organizzazione di eccellere attraverso lo sviluppo di pattern riutilizzabili e il contributo a progetti open-source?
4. In che modo l'esperto sostiene le pratiche etiche e conformi, l'automazione e l'integrazione continua nel processo di sviluppo?
5. Può raccontare i casi in cui ha proposto nuove idee e processi al settore, enfatizzando la creatività, la risoluzione dei problemi e la collaborazione interdisciplinare?



APPENDICE I RISULTATI DI APPRENDIMENTO PER L'AREA DI COMPETENZA: CREAZIONE DI CONTENUTI DIGITALI

COMPETENZA: 3.4 Programmazione



I linguaggi di programmazione comprendono un'ampia gamma di argomenti e caratteristiche che consentono agli sviluppatori di comunicare istruzioni ai computer.

I diversi linguaggi di programmazione possono enfatizzare alcuni argomenti più di altri. L'apprendimento di più linguaggi può fornire una prospettiva più ampia sulla programmazione e aiutare a scegliere lo strumento migliore per i diversi progetti.

Le applicazioni software vengono sviluppate per uno specifico dominio applicativo (ad esempio, finanza, medicina, automotive, Internet, sicurezza). La conoscenza di tale ambito è importante per comprendere, formalizzare e sviluppare le funzionalità della futura applicazione.

Lo sviluppo delle aree di conoscenza e delle attitudini per lo sviluppo di applicazioni in un linguaggio di programmazione attraverso l'istruzione formale, i programmi di formazione e l'esperienza pratica può migliorare le capacità di programmazione di un individuo, ma per operare in qualsiasi linguaggio di programmazione sono necessarie molta pratica ed esperienza.



Per sviluppare le conoscenze, le abilità e le attitudini relative alla competenza COPYRIGHT E LICENZE, diverse aree fungono da prerequisiti. Queste includono:

1. **Competenze informatiche di base:** È necessario avere familiarità con l'uso del computer, la navigazione nel file system, la creazione e la gestione di file e cartelle e l'esecuzione di operazioni di base come il copia e incolla e l'installazione di software.
2. **Matematica:** Anche se non tutti i linguaggi di programmazione richiedono una matematica avanzata, una conoscenza di base dell'aritmetica, dell'algebra e della logica può essere utile per risolvere i problemi e comprendere gli algoritmi.
3. **Pensiero logico:** La programmazione comporta la scomposizione dei problemi in piccoli passaggi logici e l'elaborazione di soluzioni. Sviluppare una forte capacità di pensiero logico vi aiuterà a scrivere codice più efficiente e organizzato.
4. **Abilità di problem solving:** La programmazione si basa sulla risoluzione dei problemi. La capacità di affrontare i problemi con metodo e di pensare in modo critico sarà essenziale per scrivere codice efficace.
5. **Conoscenza dell'inglese:** La maggior parte delle risorse di programmazione, dei tutorial e della documentazione sono disponibili in inglese. Una buona conoscenza della lingua inglese renderà più facile l'accesso al materiale didattico e la comunicazione con la comunità dei programmatori.
6. **Conoscenza di base di computer e algoritmi:** La familiarità con il modo in cui i computer elaborano i dati ed eseguono le istruzioni fornirà le basi per capire come i linguaggi di programmazione interagiscono con l'hardware. La comprensione degli algoritmi di base sarà utile anche per la programmazione.
7. **Sistemi operativi:** come Windows, MacOS o Linux. Un sistema operativo (OS) è un componente software fondamentale che gestisce l'hardware del computer e fornisce servizi e interfacce per i programmi utente. Funge da intermediario tra le applicazioni e l'hardware, consentendo agli utenti di interagire con le risorse del computer in modo più semplice ed efficiente.

AREA DI COMPETENZA: 3. CREAZIONE DI CONTENUTI DIGITALI DIMENSIONE 3: LIVELLO DI COMPETENZA		
COMPETENZA: 3.4 PROGRAMMAZIONE		
1	A livello di base e con la guida, sono in grado di farlo:	<ul style="list-style-type: none"> • identificare i principali tipi di applicazione che possono essere sviluppati in un linguaggio di programmazione, • enumerare i principali linguaggi di programmazione, • identificare cosa sia un IDE (Integrated Development Environment).
2	A livello di base e con l'autonomia e la guida appropriata dove necessario, sono in grado di:	<ul style="list-style-type: none"> • identificare le caratteristiche principali dei diversi tipi di applicazioni, • individuare le caratteristiche e le funzionalità di base dei diversi IDE, • identificare il dominio applicativo come l'area della vita reale a cui l'applicazione è destinata, • spiegare che cos'è la documentazione sulla programmazione linguistica.
3	Da solo e risolvendo problemi semplici, posso farlo:	<ul style="list-style-type: none"> • spiegare le principali specifiche/funzionalità di un'applicazione in un particolare dominio, • scegliere un linguaggio di programmazione specifico corrispondente alla piattaforma, • scegliere l'IDE giusto per uno specifico linguaggio di programmazione, • spiegare la sintassi di uno specifico linguaggio di programmazione.
4	In modo indipendente, in base alle mie esigenze e risolvendo problemi ben definiti e non routinari, sono in grado di:	<ul style="list-style-type: none"> • spiegare i tipi di dati, • spiegare e dichiarare le variabili, • spiegare come utilizzare gli operatori, • applicare le istruzioni del flusso di controllo.
5	Oltre a guidare gli altri, posso:	<ul style="list-style-type: none"> • definire, dichiarare e applicare metodi/funzioni, • eseguire operazioni di input/output (I/O), • realizzare librerie e moduli, • enumerare e spiegare i paradigmi di programmazione.
6	A livello avanzato, in base alle mie esigenze e a quelle degli altri, e in contesti complessi, sono in grado di:	<ul style="list-style-type: none"> • spiegare i concetti di programmazione imperativa orientata agli oggetti (OOP) o procedurale imperativa o logica dichiarativa o funzionale dichiarativa, • effettuare la concomitanza, • test e debug, • spiegare le differenze tra compilato e interpretato (scripting), • spiegare i concetti dei linguaggi di markup.

7	A livello altamente specializzato, posso:	<ul style="list-style-type: none"> • progetta e supervisiona l'implementazione di sistemi informatici complessi, • creare applicazioni per risolvere problemi complessi con una definizione limitata, relativi a un dominio applicativo, • integrare le mie conoscenze per contribuire alle pratiche e alle conoscenze professionali e per guidare altri nell'analisi e nello sviluppo di applicazioni.
8	Al livello più avanzato e specializzato, posso:	<ul style="list-style-type: none"> • creare soluzioni per risolvere problemi complessi con molti fattori interagenti legati a un dominio applicativo, • proporre nuove idee e processi al settore.

AREA DI COMPETENZA: 3. creazione di contenuti digitali			
COMPETENZA: 3.4 PROGRAMMAZIONE			
BASE			
Risultato dell'apprendimento	Livello	K - S - A	Esempio
1. Comprendere i tipi di applicazioni che possono essere sviluppate in un linguaggio di programmazione.	L1 - L2	K	Enumerare i principali tipi di applicazioni relative alla piattaforma di implementazione che possono essere sviluppate per desktop, mobile, web, embedded. Descrivere quali sono le principali differenze tra di esse (come possono essere utilizzate per sviluppare applicazioni per vari sistemi e ambienti).
2. Spiegare le caratteristiche dei diversi tipi di applicazioni per quanto riguarda la piattaforma e l'ambiente di lavoro.	L2	K	Enumerare le caratteristiche delle applicazioni in base alla loro piattaforma di lavoro (desktop, mobile, web, embedded) e all'ambiente (software standalone, ottimizzato per interfacce touch, accesso tramite Internet, mirato a funzioni specifiche in dispositivi che richiedono un utilizzo efficiente delle risorse e un'elaborazione in tempo reale).
3. È consapevole di come il linguaggio di programmazione possa essere utilizzato per sviluppare applicazioni per vari sistemi e ambienti.	L1 - L2	K	Riconoscere la differenza tra i linguaggi di programmazione specifici di una piattaforma e quelli multipiattaforma. Alcuni linguaggi di programmazione sono progettati specificamente per determinate piattaforme.
4. Enumerare un elenco con un linguaggio di programmazione comune.	L1	K	Conosce diversi esempi di linguaggi di programmazione: C, C++, C#, Java, Kotlin, JavaScript, HTML, XML, React, Python, Ruby, Prolog, Lisp, Scala.

5. Identificare cosa è un IDE - Ambiente di sviluppo integrato	L1	K	Sa che IDE è l'acronimo di "Integrated Development Environment". È un'applicazione software che fornisce strumenti e funzionalità complete per facilitare lo sviluppo di applicazioni software.
6. Conosce le caratteristiche e le funzionalità principali dei diversi IDE.	L1 - L2	K	Un ambiente di sviluppo integrato (IDE) offre in genere un insieme completo di caratteristiche e funzionalità per facilitare lo sviluppo del software.
7. Individuare e scegliere un IDE corrispondente a un linguaggio di programmazione	L2	S	Sa come trovare su Internet un elenco di IDE disponibili per uno specifico linguaggio di programmazione. In base alle caratteristiche degli IDE è possibile scegliere un IDE.
8. Installare un IDE	L2	S	Sa come installare un IDE utilizzando le impostazioni predefinite o personalizzate.
9. Spiegare e trovare la documentazione di programmazione del linguaggio	L1 - L2	K	Sa che esistono quattro tipi di documentazione per la programmazione linguistica: documentazione ufficiale della lingua, tutorial, post/discussioni, materiali di riferimento.
10. È in grado di eseguire un semplice programma	L2	S	Può creare, modificare e salvare file di codice sorgente con l'estensione specifica all'interno dell'IDE. È in grado di costruire e compilare il codice (solitamente tratto dalla documentazione ufficiale) all'interno dell'IDE.
11. Spiegare il controllo di versione	L2	S	Il controllo delle versioni viene utilizzato per tenere traccia delle modifiche al codice. Sa come tornare a uno stato di lavoro noto, se necessario.
12. Classificare i domini applicativi	L2	K	I domini di applicazione sono aree della vita reale a cui l'applicazione è destinata.

13. Identificare le richieste e i vincoli specifici per un determinato dominio applicativo.	L2	S	Può porre domande su considerazioni quali i requisiti dell'utente, le strutture dei dati, gli algoritmi, le misure di sicurezza e gli standard di conformità pertinenti al dominio applicativo.
---------------------------------------------------------------------------------------------	----	---	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

AREA DI COMPETENZA: 3. CREAZIONE DI CONTENUTI DIGITALI			
COMPETENZA: 3.4 PROGRAMMAZIONE			
INTERMEDIO			
Risultato dell'apprendimento	Livello	K - S - A	Esempio
14. Spiegare le principali specifiche/funzionalità di un'applicazione in un particolare dominio.	L3	K	Sa che le specifiche o le funzionalità di un'applicazione in un particolare dominio si riferiscono ai requisiti e alle capacità specifiche che l'applicazione deve avere per rispondere efficacemente alle esigenze e alle sfide di quel dominio.
15. Stabilire le principali specifiche/funzionalità di un'applicazione in un particolare dominio.	L3	A	Pone domande critiche sulle funzioni future e sui vincoli di applicazione.
16. È consapevole dei criteri di scelta di un linguaggio di programmazione	L3	K	È consapevole dei criteri di scelta del linguaggio di programmazione giusto per la propria applicazione, che dipende da diversi fattori (singola/piattaforma, web/mobile/desktop/).

corrispondente alla piattaforma di destinazione.			
17. Scegliere un linguaggio di programmazione specifico corrispondente alla piattaforma di destinazione	L3	A	Cercare informazioni sui vincoli applicativi relativi al gruppo target, alle piattaforme utilizzate. Considerare gli aspetti finanziari come gratuità/commercialità, open source, frequenza degli aggiornamenti.
18. È consapevole dei criteri di scelta di un IDE	L3-L4	K	Conosce diversi criteri per la scelta di un IDE, come l'associazione con un linguaggio di programmazione, la compatibilità con la piattaforma (singola o multiplatforma), il supporto alla programmazione, gli strumenti e i plugin integrati, l'interfaccia UX, la documentazione e la comunità, le prestazioni, la licenza e i costi.
19. Scegliere l'IDE giusto per un linguaggio di programmazione specifico	L3-L4	S	Può scegliere l'IDE in base al progetto e alle preferenze personali o del team. La scelta deve essere allineata al flusso di lavoro e ai requisiti, considerando che non esiste una soluzione unica per tutti.
20. È consapevole della definizione della sintassi di un linguaggio di programmazione.	L3	K	Sa che la sintassi, nel contesto di un linguaggio di programmazione, si riferisce all'insieme di regole e convenzioni che dettano come il codice deve essere scritto per essere considerato valido e comprensibile dal compilatore o dall'interprete.
21. È consapevole del ruolo della sintassi di un linguaggio di programmazione.	L3-L4	K	Sa che la sintassi ha sette ruoli. <ol style="list-style-type: none"> 1. Parsing del codice sorgente. 2. Rilevamento degli errori. 3. Leggibilità del codice. 4. Coerenza linguistica. 5. Regole linguistiche. 6. Evoluzione del linguaggio.

			7. Compatibilità di vari strumenti, librerie e framework.
22. È consapevole delle principali regole della sintassi di una lingua	L3-L4	K	<p>Sa che le regole principali della sintassi di un linguaggio di programmazione si riferiscono a:</p> <ol style="list-style-type: none"> 1. Commenti. 2. Terminatori e separatori. 3. Parole chiave. 4. Regole di identificazione. 5. Regole degli operatori. 6. Affermazioni ed espressioni. 7. Blocchi di codice. 8. Sensibilità al caso. 9. Regole sugli spazi bianchi.
23. È consapevole della classificazione dei tipi di dati e delle loro caratteristiche.	L4	K	<p>Sa che un tipo di dati in un linguaggio di programmazione è una classificazione che specifica il tipo di valore che una variabile o un'espressione può contenere. Definisce l'insieme delle operazioni che possono essere eseguite sui dati, la gamma di valori che possono rappresentare e lo spazio di memoria necessario per memorizzarli.</p>
24. Elencare i principali tipi di dati e le loro caratteristiche in un programma linguistico.	L4	S	<p>È in grado di elencare i principali tipi di dati, cosa rappresentano e quali sono le rappresentazioni interne ed esterne.</p>
25. È consapevole di come dichiarare una variabile	L4	K	<p>Sa che, a seconda del linguaggio di programmazione, le variabili devono essere dichiarate prima di essere utilizzate.</p> <p>Sa che un identificatore è un nome dato a un pacchetto, una classe, un'interfaccia, un metodo o una variabile.</p> <p>Conosce le regole di definizione del nome di una variabile.</p>

26. È consapevole di come utilizzare gli operatori	L4	K	<p>Sa che gli operatori in un linguaggio di programmazione sono simboli o parole chiave che rappresentano operazioni specifiche da eseguire su uno o più operandi.</p> <p>Conosce le categorie comuni di operatori.</p> <p>Conosce i tipi di operatori: unario, binario, ternario.</p> <p>Conosce la notazione degli operatori (infixo, prefixo, postfixo).</p>
27. Elencate e spiegate le istruzioni del flusso di controllo	L4	K	<p>Sa che le istruzioni di flusso di controllo sono costrutti di programmazione che determinano l'ordine di esecuzione delle istruzioni in un programma. Conosce la logica di ogni istruzione di controllo.</p>
28. È consapevole di come scrivere un semplice file sorgente.	L4	S	<p>È in grado di creare un nuovo file sorgente con l'estensione corrispondente alla lingua di programmazione.</p> <p>Sa come applicare la regola della sintassi per creare una semplice applicazione.</p>
29. È consapevole di come eseguire un file sorgente.	L4	S	<p>Sa come differenziare le fasi esatte a seconda del linguaggio di programmazione utilizzato, degli strumenti installati e dell'ambiente di sviluppo.</p> <p>È in grado di compilare per generare un file eseguibile dal codice sorgente.</p> <p>È in grado di eseguire il programma in (a) interfaccia a riga di comando (terminale), (b) interfaccia IDE.</p>
30. Implementare i metodi di gestione dei progetti	L4	S	<p>È in grado di organizzare il codice in progetti e di gestire i file di progetto. Ciò include la creazione di strutture di progetto, l'aggiunta di dipendenze e la configurazione di percorsi di compilazione.</p>

AREA DI COMPETENZA: 3. CREAZIONE DI CONTENUTI DIGITALI			
COMPETENZA: 3.4 PROGRAMMAZIONE			
AVANZATO			
Risultato dell'apprendimento	Livello	K - S - A	Esempio
31. Dichiarare metodi/funzioni	L5	K	<p>Sa che la dichiarazione di un metodo nella programmazione si riferisce al processo di definizione delle caratteristiche e del comportamento di una funzione all'interno di una classe o di un modulo.</p> <p>Sa quali sono gli elementi chiave della dichiarazione di un metodo (coerenti con la maggior parte dei linguaggi di programmazione): nome, tipo di ritorno, parametri, corpo del metodo.</p>
32. Definire metodi/funzioni	L5	K	<p>Sa che la definizione di un metodo nella programmazione si riferisce all'implementazione del comportamento, della logica e della funzionalità del metodo dichiarato in precedenza. La definizione di un metodo comprende il corpo del metodo, l'algoritmo e la logica, la dichiarazione di ritorno.</p>
33. Applicare metodi/funzioni	L5	S	<p>Sa come chiamare un metodo da varie parti del codice per eseguire le azioni specificate.</p>
34. È a conoscenza delle operazioni di input/output (I/O).	L5	K	<p>Sa che le operazioni di input/output (I/O) nella programmazione si riferiscono al processo di lettura dei dati da fonti esterne (input) o di invio dei dati a destinazioni esterne (output). Le operazioni di I/O sono essenziali per interagire con utenti, file, risorse di rete, database e altri sistemi esterni.</p>
35. Eseguire operazioni di ingresso/uscita (I/O)	L5-6	S	<p>Sa come:</p> <ul style="list-style-type: none"> - lettura di input/scrittura di output: da/verso console, file, rete. - formato di uscita. - errore di gestione. - serializzare e deserializzare.

36. È consapevole delle librerie e dei moduli	L5-6		Si riferisce al processo di utilizzo di componenti di codice pre-scritto per estendere le funzionalità di un'applicazione software. Le librerie e i moduli sono raccolte di funzioni, classi e risorse create da altri sviluppatori per fornire funzionalità specifiche.
37. Operare con librerie e moduli	L5-6	S	<p>Sa come riutilizzare il codice già scritto e testato, riducendo la duplicazione degli sforzi.</p> <p>Sa come sviluppare più velocemente sfruttando le librerie e i moduli esistenti.</p> <p>Sa come concentrarsi sulla logica di base utilizzando le soluzioni esistenti per implementare gli aspetti unici di un'applicazione.</p> <p>Sa come cercare la collaborazione della comunità che sviluppa, mantiene e aggiorna le biblioteche.</p>
38. Enumerare e spiegare i paradigmi di programmazione.	L5-6	K	<p>Sa che i paradigmi di programmazione sono approcci fondamentali alla strutturazione e all'organizzazione del codice, che definiscono come i vari componenti di un programma interagiscono tra loro. Ogni paradigma rappresenta un modo diverso di pensare alla progettazione di un programma e i diversi linguaggi di programmazione spesso supportano uno o più paradigmi.</p> <p>Sa elencare i principali paradigmi e le loro caratteristiche:</p> <ul style="list-style-type: none"> - Programmazione imperativa. - Programmazione funzionale. - Programmazione orientata agli oggetti (OOP). - Programmazione procedurale. - Programmazione logica. - Programmazione guidata dagli eventi. - Programmazione concorrente e parallela. - Programmazione strutturata.

39. Spiegare i concetti di programmazione procedurale imperativa	L6	K	Sa che la programmazione procedurale imperativa è un paradigma di programmazione che si concentra sulla specificazione di una sequenza di passi o procedure che un programma deve seguire per eseguire un compito.
40. Spiegare i concetti della programmazione orientata agli oggetti (OOP)	L6	K	Sa che la programmazione orientata agli oggetti (OOP) è un paradigma di programmazione che si concentra sull'uso di oggetti, classi e metodi per strutturare il codice e risolvere i problemi.
41. Spiegare i concetti di logica dichiarativa	L6	K	Sa che la programmazione logica dichiarativa enfatizza l'espressione di relazioni e fatti senza specificare il flusso di controllo.
42. Spiegare i concetti di funzionalità dichiarativa	L6	K	Sa che la programmazione funzionale dichiarativa si concentra sull'espressione della computazione come funzioni matematiche senza fare affidamento su stati mutabili o effetti collaterali.
43. Elencare le tecniche di concomitanza	L6	K	Sa che la concorrenza è la capacità di un programma di gestire ed eseguire più attività contemporaneamente, migliorando l'efficienza e la reattività. Elencare e spiegare le tecniche di concomitanza: - multithreading. - multiprocesso.
44. Eseguire la concomitanza	L6	S	È in grado di identificare compiti indipendenti e di determinare quali parti del programma possono essere eseguite contemporaneamente senza interferire l'una con l'altra. Può utilizzare astrazioni di concurrency di alto livello che semplificano la gestione di thread e processi.
45. Test e debug	L6	K	Sa che il collaudo e il debug sono fasi critiche dello sviluppo del software che aiutano a garantire la qualità, l'affidabilità e la correttezza di un programma. Questi processi comportano l'identificazione e la correzione di errori, difetti e comportamenti inattesi nel codice.

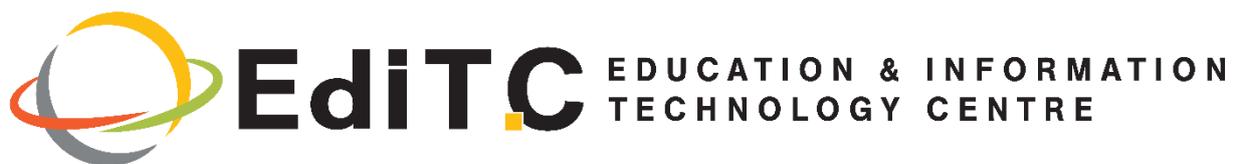
46. Debug di un'applicazione	L6	S	<p>Sa come applicare le tecniche di debug:</p> <ul style="list-style-type: none"> - Stampa delle dichiarazioni. - Debug interattivo. - Registrazione che utilizza le librerie per registrare le informazioni sull'esecuzione del programma. - Revisione del codice. - Creare un caso di test che riproduca il problema.
47. Test di un'applicazione	L7	S	<p>Sa come testare le funzionalità di un'applicazione in base alle specifiche. Sa eseguire diversi tipi di test:</p> <ul style="list-style-type: none"> - Test unitario. - Test di integrazione. - Test funzionali. - Test di regressione. - Test delle prestazioni. - Test di sicurezza.
48. Test e debug	L6	A	<p>È aperto a impegnarsi in test e debug tenendo presente che questi processi sono iterativi e richiedono attenzione ai dettagli, pazienza e approcci sistematici.</p>
49. Spiegare le differenze tra compilato e interpretato (scripting)	L6	K	<p>I linguaggi compilati e interpretati sono due approcci diversi all'esecuzione del codice nella programmazione.</p> <p>Sa che nei linguaggi compilati il codice sorgente viene trasformato in codice macchina o in una rappresentazione intermedia da un compilatore prima dell'esecuzione.</p> <p>Sa che nei linguaggi interpretati il codice sorgente viene eseguito riga per riga o dichiarazione per dichiarazione da un interprete.</p>
50. Caratteristiche del linguaggio compilato	L6	K	<p>Sa che le prestazioni di un linguaggio compilato sono migliori perché il codice viene tradotto direttamente in codice macchina, che è efficiente e ottimizzato</p>

			per l'hardware di destinazione, ma di solito è meno portabile perché è specifico dell'architettura e del sistema operativo per cui è stato compilato.
51. Caratteristiche del linguaggio interpretato	L6	S	Sa che i linguaggi interpretati hanno in genere prestazioni più lente rispetto ai linguaggi compilati a causa dell'overhead di traduzione in tempo reale, ma sono di solito più portabili in quanto il codice non è legato a una specifica piattaforma o architettura.
52. È consapevole della tipizzazione del linguaggio di programmazione	L4	K	Sa che i linguaggi di programmazione sono tipizzati staticamente o dinamicamente. Sa che nei linguaggi a tipizzazione statica i tipi di dati sono noti al momento della compilazione e il controllo dei tipi viene eseguito durante la fase di compilazione. Sa che nei linguaggi a tipizzazione dinamica i tipi di dati sono determinati in fase di esecuzione.
53. Spiegare i concetti dei linguaggi di markup	L6	K	Sa che i linguaggi di markup sono un modo di annotare il testo per definirne la struttura, la presentazione o il comportamento all'interno di un documento. Sapere che i linguaggi di markup si basano su tag/elementi di markup, attributi, nidificazione, validazione, interoperabilità tra piattaforme, dispositivi e software diversi.
54. Trasformazione e styling dei linguaggi di markup	L6	S	Sa come applicare le tecnologie di trasformazione per lo styling e la presentazione della struttura (markup) del contenuto.

AREA DI COMPETENZA: 3. CREAZIONE DI CONTENUTI DIGITALI			
COMPETENZA: 3.4 PROGRAMMAZIONE			
ALTAMENTE SPECIALIZZATO			
Risultato dell'apprendimento	Livello	K - S - A	Esempio
55. Stabilire dichiarazioni e requisiti per problemi complessi con definizione limitata.	L7	A	Porre domande critiche relative al dominio dell'applicazione per scoprire la dichiarazione del problema o i requisiti che non sono ben definiti o chiari. Collaborare con gli esperti del settore.
56. Progettare soluzioni per risolvere problemi complessi con molti fattori interagenti.	L8	S	Sa come combinare un approccio sistematico con uno strategico, come utilizzare le competenze di dominio, le abilità tecniche, l'analisi dei dati, la modellazione, la collaborazione e la volontà di adattarsi e iterare man mano che la comprensione del problema si evolve.
57. Progetta l'implementazione di sistemi informatici complessi	L7	S	È in grado di progettare e integrare componenti hardware, software e di rete per creare un sistema funzionale e sicuro. È in grado di selezionare la piattaforma di destinazione, i linguaggi o i framework per lo sviluppo, di gestire le risorse personali (tempo e competenze), considerando l'ecosistema e le librerie, le caratteristiche di sicurezza, il supporto a lungo termine e la manutenzione del linguaggio, l'integrazione con i sistemi esistenti, il budget e le licenze.
58. Supervisiona l'implementazione di sistemi informatici complessi	L7	A	È attento al fatto che le applicazioni di successo siano costruite utilizzando diversi linguaggi e tecnologie. Mantiene una mentalità aperta ed è disposto ad adattarsi al progredire del progetto e all'evoluzione dei requisiti.
59. Supervisionare la conformità alle specifiche	L7	A	È attento al rispetto delle specifiche esatte, all'implementazione delle migliori pratiche di sviluppo del codice, alle strutture di progetto, al controllo delle

			versioni e ad altri concetti che aiutano a finalizzare un'applicazione sofisticata e mantenibile.
60. Implementare i metodi di gestione dei progetti	L8	S	Sa come gestire un progetto software seguendo le fasi di pianificazione, esecuzione e consegna, rispettando i principi di gestione del team, la gestione del rischio e il controllo della qualità.
61. Guidare altri nell'analisi e nello sviluppo di applicazioni	L7	A	Integrate le vostre conoscenze e contribuite attivamente alle pratiche professionali, consentendo al vostro team e alla vostra organizzazione di eccellere nei loro progetti. Considera l'apprendimento continuo, la mentorship, i progetti collaborativi, la creazione di documentazione, lo sviluppo di modelli, lo sviluppo di standard di codifica, la progettazione di pattern riutilizzabili.
62. Proporre nuove idee e processi al settore	L8	A	Incoraggiare tutti a introdurre concetti, strategie, metodi o approcci innovativi che possano apportare cambiamenti positivi, progressi o efficienze per aiutare le organizzazioni a rimanere competitive e adattabili in un ambiente in rapida evoluzione.

Coordinatore del progetto:



Partner:



**Cofinanziato
dall'Unione europea**

Finanziato dall'Unione europea. Le opinioni espresse appartengono, tuttavia, al solo o ai soli autori e non riflettono necessariamente le opinioni dell'Unione europea o dell'Agenzia esecutiva europea per l'istruzione e la cultura (EACEA). Né l'Unione europea né l'EACEA possono esserne ritenute responsabili.