ECDL

SYLLABUS

ECDL MODUL

RAČUNALSTVO

Syllabus ver. 1.0

Syllabus ver 1.0

Cilj

Syllabus detaljno opisuje nastavni plan i program za modul Računalstvo. Kroz ishode učenja, nastavni plan opisuje znanja i vještina koje kandidat treba imati. Nastavni plan je osnovica za praktično - primijenjeni test u području ovog modula.

Izjava o odricanju odgovornosti

European Computer Driving Licence Foundation Ltd. (u daljnjem tekstu: ECDL Fondacija) uložila je najveći mogući trud kako bi ovaj dokument bio što potpuniji i točniji, ali to ne podrazumijeva nikakvo jamstvo ili obvezu. ECDL Fondacije kao izdavač nema obvezu ni odgovornost prema bio kojoj osobi ili entitetu u vezi s ikakvom štetom ili gubitkom zbog informacija sadržanih u ovoj publikaciji. ECDL Fondacija može napraviti izmjene po vlastitom nahođenju, u bilo koje vrijeme bez prethodne obavijesti.

Prijevod i prilagodba: HRVATSKI INFORMATIČKI ZBOR (HIZ), © 2017

HRVATSKI INFORMATIČKI ZBOR
10000 ZAGREB, Ilica 191e/II

Tel: +385 1 2222 722
E-mail: hiz@hiz.hr, info@ecdl.hr
URL: www.hiz.hr, www.ecdl.hr

Modul Računalstvo

Modul Računalstvo definira osnovne pojmove i vještine koje se odnose na sposobnost računalnog razmišljanja i kodiranja (programiranja) pri izradi jednostavnih računalnih programa.

Ciljevi modula

Uspješni kandidati biti će sposobni:

- Razumjeti ključne pojmove koji se odnose na računalstvo i tipične aktivnosti sadržane u izradi programa.
- Razumjeti i koristiti tehnike računalnog razmišljanja poput dekompozicije problema, prepoznavanja primjera, apstrakcije i algoritama za analizu problema i izradu rješenja.
- Izraditi, testirati i izmijeniti algoritme programa koristeći blok dijagram (dijagram tijeka) i pseudokod.
- Razumjeti ključna načela i pojmove povezane s kodiranjem i važnost dobro strukturiranog i dokumentiranog koda.
- Razumjeti i koristiti programske konstrukcije kao što su varijable, vrste podataka, te logiku programa.
- Poboljšati učinkovitost i funkcionalnost korištenjem iteracija, uvjetnih naloga, postupaka i funkcija, te događaja i naredbi u programu.
- Provjeriti i ispraviti program i osigurati udovoljavanju zahtjevima prije korištenja

1 Pojmovi računalstva	*1.1 Ključni pojmovi*	1.1.1	Definiranje pojma računalstva.
		1.1.2	Definiranje izraza računalno razmišljanje.
		1.1.3	Definirajte pojma programa.
		1.1.4	Definiranje pojma koda. Razlikovanje izvornog od strojnog koda.
		1.1.5	Razumijevanje specifikacija i opisa programa.
		1.1.6	Prepoznavanje tipičnih aktivnosti u izradi programa: analizi, dizajnu, programiranju, testiranju i poboljšanju.
		1.1.7	Razumijevanje razlika između formalnog i prirodnog jezika.
2 Računalske metode razmišljanja	*2.1 Analiza problema*	2.1.1	Prikaz tipičnih metoda koje se koriste u računalskom razmišljanju: dekompoziciji, analizi, prepoznavanju problema, apstrakciji,

			algoritmima.
		2.1.2	Korištenje metode dekompozicije kako bi se razdvojili podaci, procesi ili složeni problemi u manje dijelove.
		2.1.3	Prepoznavanje cjeline između malih razdijeljenih problema.
		2.1.4	Korištenje apstrakcije radi filtriranja nepotrebnih detalja prilikom analize problema.
		2.1.5	Razumijevanje korištenja algoritama u računalskom razmišljanju.
	2.2 Algoritmi	2.2.1	Određivanje programskih konstrukcija redosljeda niza. Opis svrhe sekvencioniranja pri projektiranju algoritama.
		2.2.2	Prepoznavanje mogućih metoda za prikazivanje problema kao što su: dijagrami tijeka, pseudokod.
		2.2.3	Prepoznavanje simbola dijagrama tijeka poput: start / stop, proces, odluka, ulaz / izlaz, konektor, strelica.
		2.2.4	Prikazivanje slijeda operacija predstavljenim dijagramom tijeka, pseudokodom.
		2.2.5	Pisanje algoritma na temelju opisa pomoću tehnike kao što su: dijagram tijeka, pseudokod.
		2.2.6	Ispravljanje grešaka u algoritmu kao što su: nedostaje programski element, netočan slijed, pogrešan ishod odluke.
3 Kodiranje	*3.1 Početak*	3.1.1	Opis značajki dobro strukturiranog i dokumentiranog koda poput: uvlačenja, odgovarajućih komentara, opisnog nazivlja.
		3.1.2	Korištenje jednostavnih aritmetičkih operatora za izvođenje izračuna u programu: +, -, /, *.
		3.1.3	Razumijevanje prednosti operatora i redoslijeda evaluacije u složenim izrazima. Razumijevanje korištenja zagrada u strukturi složenih izraza.
		3.1.4	Razumijevanje pojma parametar. Opis svrhe parametara u programu.
		3.1.5	Definiranje termina komentar. Opis svrhe komentara u programu.
		3.1.6	Primjena komentara u programu.
	3.2 Varijable i vrste podataka	3.2.1	Definiranje pojma varijable. Opis svrhe varijable u programu.
		3.2.2	Definiranje i pokretanje varijable.

		3.2.3	Dodjela vrijednosti varijabli.
		3.2.4	Korištenje imenovane varijable u programu za računanje, pohranjivanje vrijednosti.
		3.2.5	Primjena vrste podataka u programu: niz, znak, cijeli broj, float, Boolove.
		3.2.6	Korištenje agregiranih skupina vrste podataka u programu kao što su: red, popis, tuple.
		3.2.7	Korištenje ulaznih podataka korisnika u programu.
		3.2.8	Korištenje izlaznih podataka korisnika u programu.
4 Korištenje koda	*4.1 Logika*	4.1.1	Određivanje pojma logički test. Opis i svrha logičkog testa u programu.
		4.1.2	Prepoznavanje vrste Boolovih logičkih izraza kako bi se generirala istinska ili lažna vrijednost poput: =,>, <,> =, <=, <>,! =, ==, AND, OR, NOT
		4.1.3	Korištenje izraza Boolove logike u programu.
	4.2 Iteracija	4.2.1	Određivanje pojma petlje. Opis svrhe i koristi petlje u programu.
		4.2.2	Prepoznavanje vrste petlji koje se koriste za iteraciju: for, while, repeit.
		4.2.3	Korištenje iteracije (petlje) u programu kao što je: for, while, repeit.
		4.2.4	Razumijevanje pojma beskonačne petlje.
		4.2.5	Razumijevanje pojma rekurzije.
	4.3 Uvjetovanost	4.3.1	Određivanje programske konstrukcije uvjetne instrukcije. Opisivanje svrhe uvjetnih instrukcija u programu.
		4.3.2	Korištenje IF ... THEN ... ELSE uvjetnih instrukcija u programu.
	4.4 Postupci i funkcije	4.4.1	Razumijevanje pojma postupka (procedure). Opis svrhe postupka u programu..
		4.4.2	Pisanje i naziv procedure u programu.
		4.4.3	Razumijevanje pojma funkcije. Opis svrhe funkcije u programu.

		4.4.4	Pisanje i imenovanje funkcije u programu.
	4.5 Slučajevi i instrukcije	4.5.1	Razumijevanje pojma slučaj (event). Opis svrhe slučaja u programu.
		4.5.2	Korištenje uređaja kao što su: klik miša, unos tipkovnicom, klik gumba, timer.
		4.5.3	Korištenje dostupnih generičkih biblioteka kao što su: math, random, time.
5 Pokretanje, testiranje i ispravljanje grešaka	*5.1 Pokretanje programa, testiranje i ispravljanje grešaka*	5.1.1	Razumijevanje prednosti testiranja i ispravljanja grešaka u programu zbog rješavanja problema.
		5.1.2	Razumijevanje vrste grešaka u programu kao što su: sintaksa, logika.
		5.1.3	Pokretanje programa.
		5.1.4	Prepoznavanje i ispravak sintaktičkih grešaka u programu kao što su: netočan pravopis, nedostajući interpunkcijski znak.
		5.1.5	Prepoznavanje i ispravljanje logičkih grešaka u logici programa kao što su: netočan Boolov izraz, netočna vrsta podataka.
	5.2 Korištenje programa	5.2.1	Provjera programa prema početnim zahtjevima.
		5.2.2	Opis završenog programa, svrha komuniciranja i vrijednost.
		5.2.3	Utvrđivanje poboljšanja i unapređenja programa koji mogu zadovoljiti dodatne potrebe.