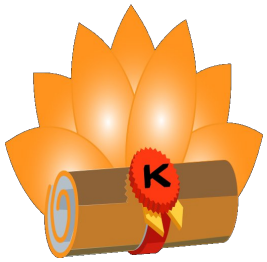
www.kamaldnp.com



KAMALDNP
kamaldnp.com
राज कमल द्वारा AI की सहायता से तैयार PDF

सॉफ़्टवेयर इंजीनियरिंग — संपूर्ण सिलेबस

BTEUP सूचना प्रौद्योगिकी में डिप्लोमा • तृतीय सेमेस्टर • सत्र 2026–27

विषय 3.3 (प्रैक्टिकम) • कुल 100 अंक • 4 क्रेडिट • L 3 – P 2

इस दस्तावेज़ के बारे में: सॉफ़्टवेयर इंजीनियरिंग सूचना प्रौद्योगिकी में डिप्लोमा के तृतीय सेमेस्टर का विषय 3.3 है। यह एक प्रैक्टिकम है (सप्ताह में 3 लेक्चर और 2 प्रैक्टिकल पीरियड) जिसके साथ 60 अंकों का थ्योरी बाह्य पेपर भी है: 40 अंक आंतरिक थ्योरी और 60 अंक बाह्य थ्योरी (3 घंटे)। इस दस्तावेज़ में आधिकारिक पाठ्यक्रम से इकाई-वार संपूर्ण सिलेबस, परीक्षा योजना, प्रैक्टिकल अभ्यास, कोर्स परिणाम और संदर्भ पुस्तकें दी गई हैं।

परीक्षा योजना

कोड	पेपर	प्रकार	L – P	क्रेडिट	आंतरिक	बाह्य	कुल
3.3	सॉफ़्टवेयर इंजीनियरिंग	प्रैक्टिकम	3 – 2	4	थ्योरी 40	थ्योरी 60 (3 घंटे)	100
सॉफ़्टवेयर इंजीनियरिंग कुल				**4**	**40**	**60**	**100**

इस विषय का उद्देश्य

सिस्टम विश्लेषण और डिज़ाइन एप्लिकेशन सॉफ़्टवेयर विकास की रीढ़ है। इस विषय को पढ़ने के बाद विद्यार्थी दी गई आवश्यकताओं के अनुसार सिस्टम विकसित और डिज़ाइन कर सकेंगे। इसमें सिस्टम के विश्लेषण और डिज़ाइन के विभिन्न चरण शामिल हैं। इसमें प्रोजेक्ट को व्यवस्थित रूप से तैयार करने का ज्ञान शामिल है। सिस्टम विश्लेषण और डिज़ाइन के विभिन्न पहलुओं को जानना महत्वपूर्ण है ताकि विद्यार्थी प्रोजेक्ट डिज़ाइन और कार्यान्वयन करते समय अपनी ज़िम्मेदारियाँ समझ सकें।

इकाई-वार संपूर्ण विषय सूची

इकाई	विषय	पीरियड	अंक (%)
1	सॉफ़्टवेयर इंजीनियरिंग का परिचय	8	20

www.kamaldnp.com

2	सॉफ़्टवेयर जीवन चक्र मॉडल	8	20
3	सॉफ़्टवेयर नियोजन, आवश्यकता विश्लेषण और विशिष्टता	10	20
4	सॉफ़्टवेयर डिज़ाइन और कार्यान्वयन	8	20
5	सॉफ़्टवेयर परीक्षण और रखरखाव	8	20

कुल: 42 पीरियड, 100%। प्रतिशत आधिकारिक सिलेबस में सुझाया गया अंक वितरण है, जो 60 अंकों के बाह्य थ्योरी पेपर में से है। सभी पाँच इकाइयों का भार बराबर है (20%), इसलिए कोई इकाई छोड़ी नहीं जा सकती।

इकाई 1 — सॉफ़्टवेयर इंजीनियरिंग का परिचय (8 पीरियड, 20%)

विषय: सिस्टम की अवधारणाएँ: सिस्टम के प्रकार (खुले, बंद, स्थिर और गतिशील सिस्टम)। सॉफ़्टवेयर इंजीनियरिंग की मूल बातें: सॉफ़्टवेयर इंजीनियरिंग की आवश्यकता — परिभाषा, प्रोग्राम बनाम सॉफ़्टवेयर उत्पाद, सॉफ़्टवेयर की विशेषताएँ।

सरल शब्दों में: सॉफ़्टवेयर इंजीनियरिंग क्या है और वह क्यों मौजूद है — सिस्टम के प्रकार जिनमें सॉफ़्टवेयर काम करता है, और सॉफ़्टवेयर उत्पाद एक अकेले प्रोग्राम से कैसे अलग है।

इकाई 2 — सॉफ़्टवेयर जीवन चक्र मॉडल (8 पीरियड, 20%)

विषय: जीवन चक्र मॉडल की आवश्यकता; क्लासिक वॉटरफ़ॉल मॉडल, प्रोटोटाइपिंग मॉडल, इवोल्यूशनरी मॉडल, स्पाइरल मॉडल, एजाइल पद्धति का परिचय।

प्रैक्टिकल अभ्यास: विभिन्न मॉडलों का तुलनात्मक विश्लेषण कीजिए।

सरल शब्दों में: सॉफ़्टवेयर प्रोजेक्ट को शुरू से अंत तक व्यवस्थित करने के विभिन्न तरीके — वॉटरफ़ॉल, प्रोटोटाइपिंग, इवोल्यूशनरी, स्पाइरल — और एजाइल का परिचय, उनकी मज़बूती और कमज़ोरी की तुलना सहित।

इकाई 3 — सॉफ़्टवेयर नियोजन, आवश्यकता विश्लेषण और विशिष्टता (10 पीरियड, 20%)

विषय: आवश्यकता संग्रह और विश्लेषण, सॉफ़्टवेयर आवश्यकता विशिष्टताएँ (SRS), अच्छे SRS की विशेषताएँ। प्रोजेक्ट आकार अनुमान के मापदंड — LOC (Lines of Code), Function Point आदि। प्रोजेक्ट अनुमान तकनीकें — COCOMO मॉडल का उपयोग। सॉफ़्टवेयर प्रोजेक्ट मैनेजर की ज़िम्मेदारियाँ।

प्रैक्टिकल अभ्यास: किसी दिए गए विषय/प्रोजेक्ट/समस्या पर SRS विकसित कीजिए।

सरल शब्दों में: कोड की एक पंक्ति लिखने से पहले प्रोजेक्ट की योजना कैसे बनती है — आवश्यकताएँ जुटाना, उन्हें SRS के रूप में लिखना, और LOC, फ़ंक्शन पॉइंट और COCOMO से आकार और प्रयास का अनुमान लगाना।

KAMALDNP.COM

www.kamaldnp.com

इकाई 4 — सॉफ़्टवेयर डिज़ाइन और कार्यान्वयन (8 पीरियड, 20%)

विषय: अच्छे सॉफ़्टवेयर डिज़ाइन की विशेषताएँ और गुण, कोहेशन और कपलिंग। सॉफ़्टवेयर डिज़ाइन दृष्टिकोण — फ़ंक्शन-उन्मुख डिज़ाइन (डेटा फ़्लो डायग्राम, डेटा डिक्शनरी, डिसीज़न ट्री और टेबल), ऑब्जेक्ट-उन्मुख डिज़ाइन। संरचित कोडिंग तकनीकें, कोडिंग शैलियाँ और दस्तावेज़ीकरण।

प्रैक्टिकल अभ्यास: समस्या का DFD मॉडल (लेवल 0 और लेवल 1 DFD) विकसित कीजिए।

सरल शब्दों में: कोड करने से पहले डिज़ाइन कैसे खींचा जाता है — डेटा फ़्लो डायग्राम, कोहेशन और कपलिंग, फ़ंक्शन-उन्मुख बनाम ऑब्जेक्ट-उन्मुख डिज़ाइन, और वे कोडिंग शैलियाँ जो कोड को पढ़ने योग्य रखती हैं।

इकाई 5 — सॉफ़्टवेयर परीक्षण और रखरखाव (8 पीरियड, 20%)

विषय: परीक्षण की अवधारणा, यूनिट परीक्षण, ब्लैक बॉक्स परीक्षण, व्हाइट बॉक्स परीक्षण, अल्फ़ा, बीटा परीक्षण, इंटीग्रेशन परीक्षण, सिस्टम परीक्षण, परीक्षण प्रकार चक्र (V-Model), सत्यापन बनाम वैधीकरण, कॉन्फ़िगरेशन प्रबंधन, टेस्ट केस का अवलोकन।

प्रैक्टिकल अभ्यास: JMeter, Canoo WebTest जैसे परीक्षण उपकरणों का उपयोग। किसी ज्ञात एप्लिकेशन के लिए टेस्ट केस लिखिए।

सरल शब्दों में: सॉफ़्टवेयर डिलीवर करने से पहले उसकी जाँच कैसे होती है — यूनिट, इंटीग्रेशन और सिस्टम परीक्षण, ब्लैक बॉक्स बनाम व्हाइट बॉक्स, V-Model, और टेस्ट केस लिखना; साथ में कॉन्फ़िगरेशन प्रबंधन, जो हर संस्करण का हिसाब रखता है।

कोर्स परिणाम

- CO1: सॉफ़्टवेयर विकास की मूल अवधारणाएँ समझना।
- CO2: जीवन चक्र मॉडलों का वर्गीकरण और उनकी तुलना करना।
- CO3: दी गई समस्या के लिए SRS लिखना और प्रोजेक्ट नियोजन अनुमान तकनीकें जानना।
- CO4: विभिन्न सॉफ़्टवेयर डिज़ाइन रणनीतियाँ समझाना।
- CO5: विभिन्न परीक्षण विधियों और परीक्षण की आवश्यकता के बारे में समझाना।

KAMALDNP • kamaldnp.com • support@kamaldnp.com
स्रोत: BTEUP IRDT कानपुर NEP-2020 पाठ्यक्रम — सूचना प्रौद्योगिकी में डिप्लोमा, सत्र 2026-27 से प्रभावी (urise.up.gov.in)। विषय-सूचियाँ आधिकारिक सिलेबस के अनुसार हैं; छोटे सारांश लेखक के अपने सरल शब्दों में लिखे गए हैं, सिलेबस का मूल पाठ नहीं हैं। कृपया अपने पॉलिटेक्निक से किसी भी स्थानीय बदलाव की पुष्टि कर लें।