

基本情報技術者 無料公開セミナー「午後試験対策学習法」

1. 基本情報技術者試験の概要と目的

◆ 試験の概要

情報処理技術者試験は、「情報処理の促進に関する法律」に基づき経済産業省が、情報処理技術者としての「知識・技能」の水準がある程度以上であることを認定している国家試験です。

情報システムを構築・運用する「技術者」から情報システムを利用する「エンドユーザ（利用者）」まで、IT に関係するすべての人に活用いただける試験として実施しています。特定の製品やソフトウェアに関する試験ではなく、情報技術の背景として知るべき原理や基礎となる技能について、幅広い知識を総合的に評価しています。

◆ 試験の目的

- ・ 情報処理技術者に目標を示し、刺激を与えることによって、その技術の向上に資すること。
- ・ 情報処理技術者として備えるべき能力についての水準を示すことにより、学校教育、職業教育、企業内教育等における教育の水準の確保に資すること。
- ・ 情報技術を利用する企業、官庁などが情報処理技術者の採用を行う際に役立つよう客観的な評価の尺度を提供し、これを通じて情報処理技術者の社会的地位の確立を図ること。

注) <http://www.jitec.ipa.go.jp> からの引用。

2. 評価と資格取得のメリット

◆ 対外的な評価

“システムの仕事に役立つ資格”を調査した「2012 年版いる資格、いない資格」(ITpro 掲載)によると、基本情報技術者は“IT ベンダーの技術者が取得すべき資格”の 3 位(1 位はプロジェクトマネージャ・2 位は応用情報技術者)に、“IT ベンダーの営業担当者が取得すべき資格”では 2 位(1 位は IT パスポート)に、また、“ユーザ企業のシステム部員が取得すべき資格”の 1 位にランキングされています。このように IT ベンダーの人事担当者の多くが、技術者やシステム部員はもちろんのこと営業担当者にも基本情報技術者を取得させたいと回答しており、システムを発注する側のユーザ企業においても取得の必要性が高い資格であることがうかがえます。

◆ 資格取得のメリット

- ① 情報処理技術者としてのスキルを有していることが公的に証明される。
- ② IT 業界で活躍する方が、現在どのレベルの専門知識を有するかが立証される。
- ③ 上級資格受験のための基礎を習得できる。
- ④ 就職活動を行う上でのアピール材料となる。
- ⑤ 資格手当や一時金など報奨金制度、昇級条件、(学生さんの場合は)履修の単位や特待生など待遇面で優遇

3. 基本情報技術者試験の出題と形式 ～何が問われるか～

(1) 午前試験

試験時間：2 時間 30 分（150 分）

出題形式：マークシートによる四肢択一形式 80 問。全問必須(1.25 点×80＝100 点)

合格基準：100 点満点の 60％ 以上の正解で合格

出題比率

テクノロジー	マネジメント	ストラテジ
50	10	20

(2) 午後試験

試験時間：2 時間 30 分（150 分）

出題形式：マークシートによる多肢選択式(複数の選択肢から 1 個以上を選択)

長文問題 13 問中 7 問を解答（次表を参照）

合格基準：100 点満点の 60％ 以上の正解で合格

午後試験の出題内容

問番号	テーマ		配点	解答数・出題数
1～4	ハードウェア	5分野から 4問を出題	各12点	5問選択/7問出題 (12×5＝60点)
	ソフトウェア			
	データベース			
	ネットワーク			
	情報セキュリティ			
5	ソフトウェア設計		各12点	5問選択/7問出題 (12×5＝60点)
6	マネジメント			
7	ストラテジ			
8	データ構造とアルゴリズム		20点	必須 (20×1＝20点)
9	C		各20点	1問選択/5問出題 (20×1＝20点)
10	COBOL			
11	Java			
12	アセンブラ			
13	表計算			

4. 平成 24 年春期の午後試験の出題テーマ・出題傾向

(1) 出題テーマ及び難易度について

問	分野	テーマ	難易度
1	ハードウェア	浮動小数点数	標準
2	ソフトウェア	コンパイラの最適化	標準
3	データベース	社員食堂の利用記録データベース の設計と運用	易
4	ネットワーク	データ転送時のフロー制御	標準
5	ソフトウェア設計	受験者数の集計リスト作成	易
6	プロジェクトマネジメント	設計工程での進捗管理	標準
7	経営・関連法規	正味現在価値による投資採算性 の評価	難
8	データ構造及びアルゴリズム	ビットの検査	標準
9	ソフトウェア開発(C)	会議時間の調整	標準
10	ソフトウェア開発(COBOL)	遊園地の入園者情報の集計	標準
11	ソフトウェア開発(Java)	試験の成績管理	標準
12	ソフトウェア開発(アセンブラ)	数字列の加算	標準
13	ソフトウェア開発(表計算)	図書管理及び図書推薦	難

(2) 出題傾向について

選択問題のブロック(前半 7 問)では、テクノロジ系・マネジメント系に解きやすい問題が多く、ストラテジ系の難易度が高めです。また計算問題が 7 問中 4 問出題され、計算の比重が高いといえます。

必須問題のアルゴリズムについては、ビット処理が取り上げられました。ビット処理は過去にも出題されているテーマですが、今回のビット処理の問題はプログラムの穴埋め、トレース、処理効率を問うバランスの良い内容で、かつ解き易い問題と思われます。

ソフトウェア開発(プログラム言語)では、表計算を除くと、言語間の難易度に差が見られず、FE の言語問題としては、このレベルが要求されているものと考えられるべきでしょう。一方、表計算は、仕様の改訂によって出題範囲が広がり、従来よりもやや難易度を上げた出題が予想されていたのですが、今回を見る限り、予想を上回る難易度の高い問題でした。

午後全体としては、前回(23 年秋)よりも、やや難易度が上がったと考えます。

(3) 出題内容及び難易度について

● 共通問題

問題番号	内 容
問 1 (ハードウェア)	「浮動小数点数」の計算問題です(19 年秋に類似の出題あり)。演算時のけた合わせや正規化などが論点になっています。類似の演習経験があれば、手順に従って計算することで答えられる問題です。
問 2 (ソフトウェア)	午後では初出題の「コンパイラの最適化技法」です。疑似言語プログラムがいくつか提示されており、これがどの最適化技法に該当するかなどが問われています。一見すると難しそうですが、実際に解いてみると意外と解き進めることができるでしょう。
問 3(データベース)	SQL 文を中心とした、オーソドックスで解きやすい問題です。
問 4 (ネットワーク)	「フロー制御」を扱った計算問題です。後半の設問ではパケットの転送時間とバッファサイズの関係性を考察して答える必要があり、見かけよりも手間がかかる問題です。
問 5 (ソフトウェア設計)	「ファイル処理」に関するオーソドックスな問題です。ソフトウェア設計は、近年の出題傾向から「プロセス中心設計(ファイル処理など)」と「オブジェクト指向(UML)」の二つを柱に置き、対策すべきでしょう。
問 6 (プロジェクト マネジメント)	設計工程での「進捗管理」に関する計算問題です。ここ数回のマネジメント系の中では比較的解きやすく、内容さえ正しく理解できれば、空欄の多くに答えることができます。
問 7 (経営戦略)	「正味現在価値(NPV)」を扱った計算問題です。内容が理解しにくい上に設問ごとに提示されている求め方に従って図表の数値を用いて計算しなければならず、非常に手間がかかります。時間的難易度が高い問題といえるでしょう。

● 必須問題

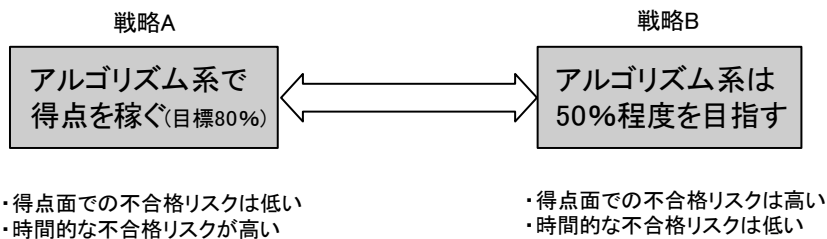
問 8 (データ構造と アルゴリズム)	今回は「ビットの検査」が出題されています。プログラムの穴埋めやトレースに加え、プログラム中の処理量の値を使って、実行回数分を計算し“処理効率”を答える、といった内容です。問題文、プログラムとも読みやすく、近年のアルゴリズム問題の中では、比較的解きやすいでしょう。 なお、新試験からのアルゴリズムでは「トレース」が重視されており、今回の設問 2 や設問 3 でもトレースで確認しないと、誤った答えを選んでしまうかもしれません。アルゴリズムの基礎知識とともに、演習で経験を積んでおくべきでしょう。
---------------------------	---

● プログラミング言語

問 9(C)	構造体を用いた C としてはオーソドックスな問題です。ただし、条件がやや複雑なため、全体の処理の把握に時間を要するかもしれません。
問 10(COBOL)	集計処理が主テーマです。レコードの内容に関する説明が多く、プログラムで何をすべきかを適確に把握しておく必要があります。
問 11(Java)	多くのコレクションが登場しており、コレクションの知識がないと厳しいでしょう。コレクションの知識があれば、比較的解答しやすい問題です。
問 12(アセンブラ)	数字列の加算処理の問題です。前半は加算に伴うけた上げに注意すれば、それほど苦戦することはありませんが、設問 3 のプログラムでは、レジスタに設定されるアドレスなどについて考察する必要があります。
問 13(表計算)	問題ボリュームが多めで、マクロが 2 つ出題されています。特に後半のマクロでは数学的な知識が要求されており、目的の値の求め方が把握しづらいなど、時間的難易度の高い問題です。

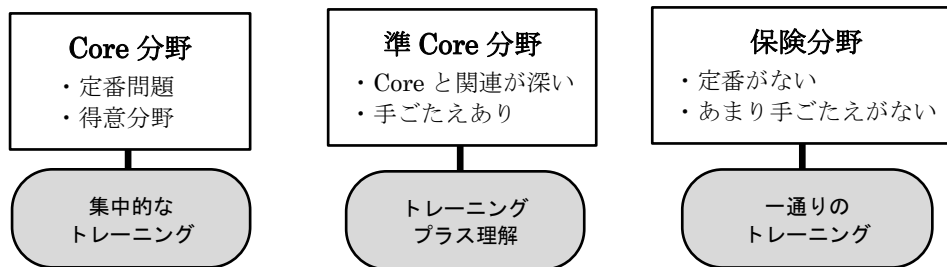
5. 午後対策 分野別学習法

(計画しておきたいこと)

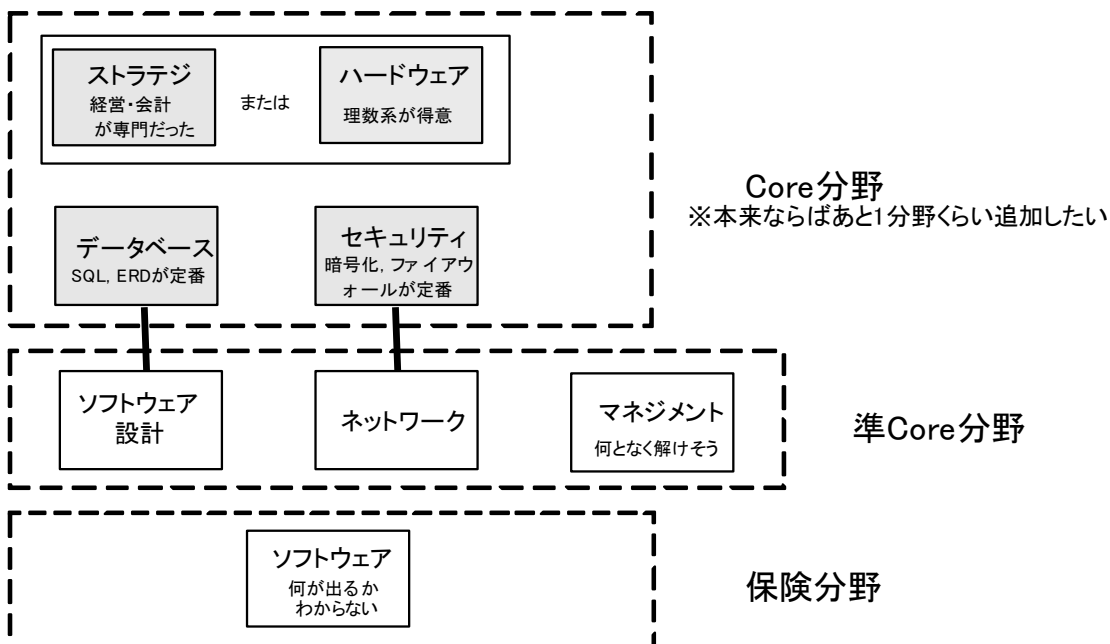


(1) 午後 選択問題対策

- ・トレーニング中心。間違えてしまった設問は解説で理解する。
場合によってはテキストに戻って復習し直す。
- ・原則として苦手な分野は選ばない。(午前と同様)分野にメリハリを付ける。
- ・Core 分野, 準 Core 分野, 保険分野に分ける
 - Core 分野は強力な得点源にする。
 - 準 Core 分野は Core と同じくらいに育てる
 - 保険分野はそのもの保険, 余力を入れない

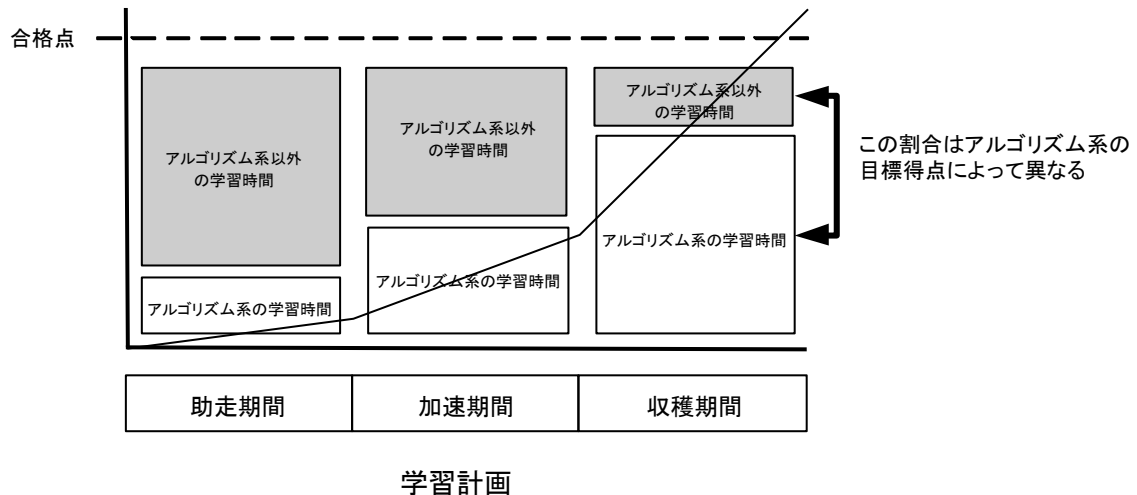


【選択分野の例】



(2) 午後 アルゴリズム系の対策

午後対策の学習スケジュールは、アルゴリズム系の対策が鍵！



★助走期間(0.5～2 ヶ月)

- ・プログラムの全体像や流れをつかむようにする。
- ・アルゴリズムのテキストを眺めてみる。アルゴリズムの読解を「試みる」。
→ 他分野の学習を間に入れながら学習する。

★加速期間(1～2 ヶ月)

- ・テキストをしっかりと読んで、疑問点をつぶしてゆく。
→ 文法や記述法などを確実に覚えてゆく。
→ わからないければ先に進む(先に進むことで、後から解決することもある)。
- ・この期間ではできるだけアルゴリズムやプログラミング言語に集中する。

★収穫期間(1～2 ヶ月)

- ・過去問題 or 問題集などに収録された問題をとにかく解く。
- ・わからないところは、解説などを読んで理解する。
→ とにかく「理解する」まではプログラムに取り組む。
- ・初めのうちは解く時間にはこだわらない。終盤は解答時間(1 題あたり 20～25 分)を意識したタイムトレーニングを行う。

6. 午後の解法テクニックを伝授

～ アルゴリズム問題の攻略法 ～

「アルゴリズム(擬似言語)」は、基本情報技術者試験の午後問 8 に出題される必須問題です。配点が高く、午後試験をクリアする上での「関門」といってもよいでしょう。アルゴリズムを苦手とする多くの方が「どのように解けばよいかわからない」と感じておられるようです。本セミナーでは、TAC 講師によるアルゴリズム問題へのアプローチや解答テクニックを一部公開したものです。もちろん、“これらがすべてではありません” が、アルゴリズムが苦手という方は、ここに紹介する方法を試してみることを、お勧めします。

(1) 擬似言語のアプローチ

擬似言語を解くとき、講師などの経験者は、一体どのようなアプローチをとっているのでしょうか。実は、経験者がとるアプローチは「分割して考える」ことなのです。

プログラム全体を一度に考えようとすると、難しく感じるでしょう。プログラムをいくつかのブロックに分割してみると、それぞれのブロック単位では、構造が比較的単純であることに気がつきます。プログラムを分割することにより、読解を容易にし、これを積み上げていくことで、プログラム全体を把握できる場合が多いのです。

(2) ブロック分割

プログラムは分割して考える、とはいえ、やみくもに分割しては効果がありません。意味のあるブロックに分割することが重要なのです。慣れないうちは、プログラムの制御構造に従って分割すればよいでしょう。(午後対策講義・演習では、より具体的なテクニックをお教えします)

(3) 変数の役割を明らかにする

プログラムで用いる変数の役割を明らかにしましょう。ブロック内だけで明らかにならない場合は、プログラム全体に視点を広げましょう。プログラムからその変数を用いている命令を抜き出せば、だいたいの想像がつくはずです。特に変数の多いプログラムでは、多少時間がかかっても、変数の役割をまとめておくことが必要です。また、添字などは図示するとよいでしょう。

(4) トレースは最小限に抑える

アルゴリズムを理解するためには「トレース」がかかせません。丁寧にトレースすれば、プログラムの動作が理解できるのです。ただし、本試験では、トレースは「必要最小限」に抑えなければなりません。ここでは思い切って、

“練習ではベタトレース推奨、ただし、本試験ではトレースは必要最小限”

と言いたいと思います。こうすることで、本試験で時間のロスを防ぎ、効率的に解くことができるようになります。(具体的なテクニックについては、講義・演習でお教えいたします)

(5) 基本部品を蓄積する

プログラムの定型的な処理パターン(基本部品)を覚えておきましょう。

8. TAC の午後対策 学習の流れ(午後対策本科生の例)

● 講義

アルゴリズムをはじめとする午後分野の知識習得には時間がかかります。したがって、早い時期から体系的に学習していくことが大切です。

- ・ アルゴリズム・データ構造の知識を習得。 ※ プログラミング言語も同様
- ・ 基本問題の演習（問題を解くことで知識のモレ・弱点を確認）を行う。
- ・ 弱点箇所、未履修のテーマを必ず再確認する → 苦手分野を少なくする。

● 演習

午後対策の専用教材を使って、答案練習を行います。

- ・ 広い範囲の過去・模擬問題の演習
- ・ 擬似言語・プログラミング言語など午後の応用演習
- ・ 選択問題対策

TAC式 解法テクニック

● 公開模試で腕試し

試験直前期の実力診断・弱点分野の把握に最適です。

※ 基本情報技術者の午後試験では「プログラミング言語」が出題されます。

ご自身の学習目的にあったプログラミング言語を選択しましょう。

プログラミング言語の種類・特徴

名 称	特 徴	学習時間	過去の難易度
C 言語	システム記述からアプリケーション作成まで広く用いられる開発者向けの言語です。	やや多め	標準～やや難
COBOL	英文形式で命令を記述するため、理解しやすい事務処理用の言語ですが、現在ではやや古さを感じる面もあります。	普通	やや易～標準
Java	最新のオブジェクト指向型言語で広く利用されており、人気の高い言語です。多機能な反面、学習内容は多くなります。	多め	標準～やや難
アセンブラ CASL II	試験用のアセンブラ言語です。命令数が少なく、理解し易い言語ですが、試験用言語なので実務向きではありません。初学者(開発系)向き。	やや少なめ	やや易～標準
表計算	試験唯一のユーザ向き言語です。身近で理解し易く、実務にも活かせます。初学者(ユーザ系/文系出身者)向き。	少なめ	やや易～やや難