

# 目次

<b>第 1 章</b>	<b>画面への出力とキーボードからの入力</b>	<b>1</b>
1-1	<b>C++ の歴史</b> . . . . .	<b>2</b>
1-2	<b>まずは画面に表示</b> . . . . .	<b>4</b>
	■ コンソール画面への出力	4
	■ ソースプログラムとソースファイル	4
	■ プログラムの実行	5
	■ コメント (注釈)	6
	■ ヘッダとインクルード	6
	■ std 名前空間の利用	7
	■ コンソール画面への出力とストリーム	8
	■ 文字列リテラル	8
	■ 改行	8
	■ main 関数	9
	■ 文	9
	■ ストリームへの連続した出力	10
	■ 警告	10
	■ 記号文字の読み方	11
	■ 自由形式記述	12
	■ インデントのすすめ	13
1-3	<b>変数</b> . . . . .	<b>14</b>
	■ 演算結果の出力	14
	■ 整数リテラル	14
	■ 演算結果の出力	14
	■ 変数	15
	■ 変数の宣言	15
	■ 代入演算子	16
	■ 変数の値の表示	17
	■ 算術演算子と演算のグループ化	17
	■ 変数と初期化	18
	■ 初期化を伴う宣言	18
	■ 初期化と代入	19
1-4	<b>キーボードからの入力</b> . . . . .	<b>20</b>
	■ キーボードからの入力	20
	■ 演算子とオペランド	21
	■ 連続した読み込み	22
	■ 単項の算術演算子	23
	■ 実数値の読み込み	24
	■ 乱数の生成	26
	■ 文字の読み込み	28
	■ 文字列の読み込み	28
	<b>まとめ</b> . . . . .	<b>30</b>

## 第 2 章 プログラムの流れの分岐 33

2-1	if 文	34
	■ if 文 (その 1)	34
	■ 関係演算子	35
	■ if 文 (その 2)	36
	■ 等価演算子	38
	■ 論理否定演算子	39
	■ 入れ子となった if 文	40
	■ 式と評価	42
	■ 式	42
	■ 評価	42
	■ 式文と空文	43
	■ 論理演算子	44
	■ 論理積演算子 &&	45
	■ 論理和演算子	45
	■ 二択条件演算子	46
	■ 二択条件演算子	46
	■ 三値の最大値	48
	■ ブロック	50
	■ コンマ演算子	52
	■ 二値のソート	52
	■ 条件における宣言	54
	■ 実数の剰余	54
2-2	switch 文	56
	■ switch 文	56
	■ ラベルと break 文	58
	■ if 文と switch 文	59
	■ 選択文	59
2-3	キーワード・識別子・演算子	60
	■ キーワード	60
	■ 識別子	61
	■ 区切り子	62
	■ リテラル	62
	■ 演算子	62
	■ 優先度	62
	■ 結合規則	62
	■ 代入式の評価	63
	まとめ	66

## 第 3 章 プログラムの流れの繰返し 69

3-1	do 文	70
	■ do 文	70
	■ 一定範囲の値の読み込み	72

	■ 数当てゲーム	72
<b>3-2</b>	<b>while 文</b>	<b>74</b>
	■ while 文	74
	■ 増加演算子と減少演算子	75
	■ 後置増加演算子と後置減少演算子	75
	■ 前置増加演算子と前置減少演算子	77
	■ 左辺値と右辺値	77
	■ 文字リテラル	78
	■ while 文と do 文	79
	■ 複合代入演算子	80
	■ 数値の逆転	80
	■ 整数の和を求める	82
<b>3-3</b>	<b>for 文</b>	<b>84</b>
	■ for 文	84
	■ 繰り返し文	86
<b>3-4</b>	<b>多重ループ</b>	<b>88</b>
	■ 九九の表	88
	■ setw 操作子	88
	■ 直角三角形の表示	90
<b>3-5</b>	<b>break 文と continue 文と goto 文</b>	<b>92</b>
	■ break 文	92
	■ continue 文	94
	■ goto 文	96
<b>3-6</b>	<b>拡張表記と操作子</b>	<b>98</b>
	■ 拡張表記	98
	■ 三つ組と二つ組	101
	■ 操作子	102
	<b>まとめ</b>	<b>104</b>

---

## 第 4 章 データ型と配列 107

<b>4-1</b>	<b>整数型</b>	<b>108</b>
	■ 整数型	108
	■ <climits> ヘッダ	110
	■ 文字型	112
	■ 符号付き整数型と符号なし整数型	116
	■ 整数リテラル	118
	■ オブジェクトと sizeof 演算子	120
	■ size_t 型と typedef	122
	■ typeid 演算子	123
	■ bool 型	124
<b>4-2</b>	<b>浮動小数点型</b>	<b>126</b>
	■ 浮動小数点型	126
	■ 浮動小数点リテラル	127

<b>4-3</b>	<b>演算と型</b> . . . . .	<b>130</b>
	■ 演算と型	130
	■ 明示的型変換	132
	■ キャスト記法による明示的型変換	132
	■ 関数的記法による明示的型変換	133
	■ static_cast 演算子による明示的型変換	134
	■ 繰返しの制御	136
<b>4-4</b>	<b>配列</b> . . . . .	<b>140</b>
	■ 配列	140
	■ 配列と for 文	142
	■ 配列による成績処理	144
	■ 定値オブジェクト	144
	■ 配列の初期化	146
	■ 配列の要素数	147
	■ 配列型の情報	148
	■ 配列を逆順に並べかえる	148
	■ 配列のコピー	150
	■ 多次元配列	152
	■ 初期化子	154
<b>4-5</b>	<b>列挙体</b> . . . . .	<b>156</b>
	■ 列挙体	156
	■ 列挙子の値	158
	■ 名前のない列挙体	158
	■ 列挙体を利用するメリット	158
	<b>まとめ</b> . . . . .	<b>160</b>

---

## 第 5 章 関数の基本 163

<b>5-1</b>	<b>関数とは</b> . . . . .	<b>164</b>
	■ 関数	164
	■ 飛越し文	168
	■ main 関数	168
	■ 関数原型宣言	170
	■ 値渡し	172
	■ 引数を受け取らない関数	174
	■ 他の関数を呼び出す	175
	■ void 関数	176
	■ 関数の汎用性	177
	■ 省略時実引数と引数の型	178
<b>5-2</b>	<b>参照と参照渡し</b> . . . . .	<b>180</b>
	■ 値渡しの限界	180
	■ 参照	181
	■ 参照渡し	182
	■ 値渡しと参照渡し	182

	■ 三値のソート	184
<b>5-3</b>	<b>整数の内部を調べてみよう</b>	<b>186</b>
	■ 整数の内部	186
	■ 符号なし整数の内部表現	186
	■ 符号付き整数の内部表現	187
	■ ビット単位の論理演算	190
	■ シフト演算	192
	■ ビット数のカウント	194
	■ unsigned 型のビット数の取得	194
	■ ビット内容の表示	195
<b>5-4</b>	<b>有効範囲と記憶域期間</b>	<b>196</b>
	■ 有効範囲	196
	■ 記憶域期間	198
	■ 参照を返す関数	202
<b>5-5</b>	<b>多重定義とインライン関数</b>	<b>204</b>
	■ 関数の多重定義	204
	■ インライン関数	206
	<b>まとめ</b>	<b>210</b>

## 第 6 章 ポインタ 213

<b>6-1</b>	<b>ポインタ</b>	<b>214</b>
	■ オブジェクトとアドレス	214
	■ アドレス演算子によるアドレスの取得	214
	■ ポインタ	216
	■ アドレス演算子によるポインタの生成	216
	■ 間接参照演算子	217
<b>6-2</b>	<b>関数呼出しとポインタ</b>	<b>220</b>
	■ 関数呼出しと値渡し	220
	■ ポインタの値渡し	222
<b>6-3</b>	<b>ポインタと配列</b>	<b>224</b>
	■ ポインタと配列	224
	■ 配列名と先頭要素へのポインタ	225
	■ 関数間の配列の受渡し	226
	■ const ポインタ型の仮引数	228
<b>6-4</b>	<b>オブジェクトの動的な生成</b>	<b>230</b>
	■ 自動記憶域期間と静的記憶域期間	230
	■ 動的記憶域期間	230
	■ オブジェクトの初期化	233
	■ 配列オブジェクトの動的生成	234
	■ オブジェクトの生成に失敗したとき	236
	■ 空ポインタ	238
	■ void へのポインタ	239
	<b>まとめ</b>	<b>240</b>

## 第 7 章 文字列とポインタ 243

<b>7-1</b>	<b>文字列とポインタ</b> . . . . .	<b>244</b>
	■ 文字列リテラル	244
	■ 文字列	246
	■ 文字列の初期化	246
	■ 空文字列	247
	■ 文字列の読み込み	248
	■ 関数間の変渡し	249
	■ ポインタによる文字列の表現	250
	■ ポインタによる文字列が好まれる理由	251
	■ 文字列の配列	252
<b>7-2</b>	<b>cstring ライブラリ</b> . . . . .	<b>254</b>
	■ 文字列の長さ：strlen	254
	■ 文字列のコピー：strcpy と strncpy	256
	■ 文字列の連結：strcat と strncat	257
	■ ポインタを返却する関数	258
	<b>まとめ</b> . . . . .	<b>262</b>

## 第 8 章 関数の応用 265

<b>8-1</b>	<b>再帰的な関数</b> . . . . .	<b>266</b>
	■ 再帰とは	266
	■ 階乗値	267
	■ 再帰呼出し	268
<b>8-2</b>	<b>関数テンプレート</b> . . . . .	<b>270</b>
	■ 関数テンプレートとテンプレート関数	270
	■ 明示的な具現化と特殊化	274
<b>8-3</b>	<b>規模の大きなプログラムの開発</b> . . . . .	<b>276</b>
	■ 分割コンパイルと結合	276
	■ 結合	277
	■ ヘッダの作成	280
<b>8-4</b>	<b>名前空間</b> . . . . .	<b>282</b>
	■ 名前空間の定義	282
	■ using 宣言と using 指令	284
	■ using 宣言	284
	■ using 指令	284
	■ 名前なし名前空間	285
	■ 別名の定義	285
<b>8-5</b>	<b>関数へのポインタと動的な関数呼出し</b> . . . . .	<b>286</b>
	■ 関数へのポインタ	286
	■ 関数へのポインタ	287
	■ メニュードリブンプログラム	290
	<b>まとめ</b> . . . . .	<b>292</b>

## 第 9 章 クラスの基本 295

9-1	クラスの考え方	296
	■ データの扱い	296
	■ オブジェクトとクラス	297
	■ クラス定義	298
	■ クラス型のオブジェクト	299
	■ メンバのアクセス	300
	■ 問題点	300
	■ 非公開メンバと公開メンバ	302
	■ コンストラクタ	303
	■ メンバ関数	303
	■ メソッドとメッセージ	304
	■ クラスとオブジェクト	305
9-2	クラスの実装	306
	■ クラス定義の外部でのメンバ関数の定義	306
	■ インタフェース部と実装部の分離	308
	■ -> 演算子によるメンバのアクセス	310
	■ 自動車クラス	311
	まとめ	316

## 第 10 章 単純なクラスの作成 319

10-1	日付クラスの作成	320
	■ 日付クラス	320
	■ コンストラクタ	320
	■ 初期化とコンストラクタ	322
	■ 代入	324
	■ デフォルトコンストラクタ	326
	■ const メンバ関数	330
	■ 文字列ストリーム	333
	■ 挿入子と抽出子の定義	334
10-2	メンバとしてのクラス	336
	■ クラス型のメンバ	336
	■ has-A	336
	■ コンストラクタ初期化子	338
	■ ヘッダの設計とインクルードガード	341
	■ クラス型の引数	344
	■ this ポインタ	346
	■ メンバへのポインタ	348
	まとめ	350

## 第11章 演算子の多重定義 353

<b>11-1</b>	<b>カウンタクラス</b> . . . . .	<b>354</b>
	■ カウンタクラス	354
	■ カウンタに対する操作	356
	■ 変換関数	356
	■ 演算子の多重定義	357
	■ 論理否定演算子	358
	■ 増加演算子と減少演算子	358
<b>11-2</b>	<b>真理値クラス</b> . . . . .	<b>362</b>
	■ 真理値クラス	362
	■ クラス有効範囲	362
	■ 変換コンストラクタ	363
	■ ユーザ定義変換	364
	■ 変換関数	364
<b>11-3</b>	<b>複素数クラス</b> . . . . .	<b>366</b>
	■ 複素数	366
	■ 演算子関数とオペランドの型	366
	■ フレンド関数	368
	■ const 参照引数	370
	■ 効率の向上	372
	■ 複合代入演算子	372
	■ 等価演算子	373
	■ 単項算術演算子	373
	■ 演算子関数に関する規則	376
	<b>まとめ</b> . . . . .	<b>380</b>

## 第12章 配列クラスで学ぶクラス的设计 383

<b>12-1</b>	<b>デストラクタ</b> . . . . .	<b>384</b>
	■ 整数配列クラス	384
	■ 明示的コンストラクタ	384
	■ 添字演算子	385
	■ クラスオブジェクトの生存期間	386
	■ デストラクタ	388
<b>12-2</b>	<b>代入演算子とコピーコンストラクタ</b> . . . . .	<b>390</b>
	■ 代入演算子の多重定義	390
	■ コピーコンストラクタ	394
<b>12-3</b>	<b>例外処理</b> . . . . .	<b>396</b>
	■ エラーに対する対処	396
	■ 例外処理	397
	■ 例外の捕捉	398
	■ 例外の送付	400



12-4 静的メンバ	404
■ 静的データメンバ	404
■ 静的メンバ関数	406
■ 日付クラス	408
まとめ	410

## 第13章 クラステンプレート 413

13-1 配列クラス	414
■ クラステンプレート	414
■ クラステンプレート	416
■ メンバ関数の実装	416
■ テンプレートクラス	416
13-2 スタッククラス	418
■ スタック	418
■ スタックとは	418
■ スタックと関数呼出し	418
■ スタックの実現	420
■ コピーコンストラクタと代入演算子の無効化	421
■ 利用例	424
13-3 ベクトル vector	426
■ コンテナと vector	426
■ 構築	427
■ 容量と要素数	428
■ 要素のアクセス	430
13-4 文字列 string	434
■ string クラス	434
まとめ	440

## 第14章 継承 441

14-1 継承	442
■ 元号形式の日付クラス	442
■ 継承	446
■ 派生グラフ	447
■ 派生の形態	448
■ private 派生	448
■ protected 派生	450
■ public 派生	451
■ 三つの派生のまとめ	452
■ 継承による日付クラス	452
■ コンストラクタ	454
■ is-A	457
■ コンストラクタとコンストラクタ初期化子	458

14-2	偶数カウンタクラス	460
	■ 偶数カウンタ	460
	■ インタフェースの制限	460
	■ using 宣言によるアクセス権の調整	461
	■ 同一名メンバ関数の再定義	461
	まとめ	464

## 第15章 仮想関数 467

15-1	継承の問題点	468
	■ ペットクラス	468
	■ ペットを指す配列	470
	■ 継承版のペットクラス	471
	■ 基底クラスへのポインタと派生クラスへのポインタ	474
	■ 基底クラスへのポインタに対するメンバ関数呼出し	476
15-2	仮想関数と多相的クラス	478
	■ 仮想関数	478
	■ 多相的クラス	480
	■ オーバライド	480
	■ 仮想関数と呼出しメカニズム	481
	■ 仮想デストラクタ	482
15-3	実行時型情報とキャスト	484
	■ 実行時型情報	484
	■ 動的キャスト	486
	■ ダウンキャスト	488
15-4	線形リスト	490
	■ 線形リスト	490
	■ ノードクラス	490
	■ 線形リストクラス	491
	■ フレンドクラス	493
	まとめ	502

## 第16章 抽象クラス 505

16-1	抽象クラス	506
	■ 図形クラス	506
	■ 抽象クラスと純粋仮想関数	508
	■ 純粋仮想関数	508
	■ 抽象クラス	509
16-2	抽象クラステンプレートによるスタック	510
	■ スタッククラス	510
	まとめ	516

<b>第17章</b>	<b>多重継承</b>	<b>517</b>
<b>17-1</b>	<b>多重継承</b>	<b>518</b>
	■ 多重継承	518
	■ 基底クラスの初期化と曖昧さの制御	519
	■ 基底クラスの初期化	520
	■ 曖昧さの解決	521
	■ クロスキャスト	522
<b>17-2</b>	<b>仮想基底クラス</b>	<b>526</b>
	■ 仮想基底クラス	526
	■ 初期化	528
	<b>まとめ</b>	<b>530</b>
	<b>参考文献</b>	<b>531</b>
	<b>索引</b>	<b>533</b>
	<b>謝辞</b>	<b>547</b>
	<b>著者紹介</b>	<b>548</b>