

```

    case 2 : C = 0.5088 - 0.0011*(B-135); break;
    case 3 : C = 0.4978 - 0.0010*(B-145); break;
    case 4 : C = 0.4878 - 0.0009*(B-155); break;
    default : printf("yanlıř seçenek\n");}
cout << setprecision (4);
// 4 hane sayı yazılacağı belirtilir
cout << " Kazanan sporcu= " << setw (4) << B << endl;
cout << setprecision (5); //5 haneli sayı yazılacağı belirtilir
cout << " ve C değeri = " << setw (5) << C << endl;
system("PAUSE");
return 0;}

```

3.5 Dosyalar (6. Hafta)

Bilgisayar tarafından değerdendirilen bilgiler daha sonra kullanılacaksa kayıt edilmeleri daha uygun olacaktır. Kayıt edilen verilerin kullanımı için dosyalar kullanılır. Dosyalar FILE komutu ile pointer tanımlayıcısı kullanılarak ifade edilirler. Yazacağınız programa dosyaları kullanmadan önce FILE *fin, *fout; işaretçisi (pointer) ile tanımlanmalıdır (burada * işaretleri işaretçiyi, fin ve fout dosya isimlerini tanımlar). Örnek olarak dosya sadece okuma amaçlı açılacak ise dosya için fin=fopen("dosya1.dat","r"); kullanılabilir. Bu ifade ile dosya1.dat isimli veri dosyası sadece içindeki veriler okunması (r=reading) için açılmaktadır. Sadece yazma amaçlı dosya açılmak istenirse fout=fopen("dosya1.dat","w"); (w=write) ifadesi kullanılabilir. Dosyalarla ilgili işlemler bittikten sonra dosyaların kapatılması uygun olacaktır. Bunun için, fclose(fin); fclose(out); deyimleri kullanılır. Klavyeden veri giriři için kullanılan scanf komutu yerine dosyadan veri okumada fscanff, ekrana veri yazma işleminde kullanılan printf komutu yerine ise fprintf kullanılır.

Dosyayı, (ANSI C) programa tanıtmak için fopen(dosyadi, mod); kullanılır. Buradaki mod dosyanın açılıř biçimini belirtir ("r"-read-sadece okuma, "w"-write-sadece yazma, "a"-add-ekleme). Ařağıdaki programda okuma amaçlı açılan dosyadan veri okuması yapılmaktadır.

C++ da dosyadan okuma ve yazma işlemleri için kullanılacak içerik dosyalarının kullanılması gerekmektedir. Bunlar:

ifstream.h – dosyadan okuma, dosyaya giriş işlemleri için kullanılır

ofstream.h – dosyaya yazma, dosyadan çıkış işlemleri için kullanılır

fstream.h – dosyadan okuma, dosyaya yazma işlemleri için kullanılır

Örnek 3-19 Ařağıdaki C++ programında,fstream dosya1("c:\\veri.txt", ios::out); komutu ile açılan metin dosyasına veriler yazdırılmakta (dosya1 << "Test Write to file"); ve işlem bitince dosya kapatılmakta (dosya1.close();) ve program sonlanmaktadır.

Program 3-15 Dosyaya veri yazan bir C++ programı.

```
#include <fstream.h>
int main()
{
    fstream dosyal("c:\\veri.txt", ios::out);
    dosyal << "Fizik Mühendisliği Bölümü";
    dosyal.close();
    return 0;
}
```

Örnek 3-20 Bir bankadan çekilen A miktarındaki kredinin aylık geri ödeme tutarı P'nin ne kadar olacağını hesaplayan algoritma ve program aşağıda verilmektedir. Geri ödeme planında aylık faizin (r) ve ödeme süresinin (n) bilinmesi gerekmektedir. Sonuçlar bir dosyaya yazdırılmaktadır. Faiz hesabı formülü: $P=A \times r / (1 - (1+r)^{-n})$.

Algoritma 3-12 Kredi faiz hesabı.

1. Başla
2. Çekilecek kredi miktarını giriniz, A
3. Aylık faiz değerleri, r1, r2, r3, r4, r5 (1, 2, 3, 5 ve 10 yıl)
4. Geri ödeme P1, P2, P3, P4, P5 (1, 2, 3, 5 ve 10 yıl),
5. $n1=1 \times 12$,
6. $P1=A \times r1 / (1 - (1+r1)^{-n1})$,
7. A, r1, n1, P1 ve $P1 \times n1$ değişkenlerindeki değerleri dosyaya yaz,
8. $n2=2 \times 12$,
9. $P2=A \times r2 / (1 - (1+r2)^{-n2})$,
10. A, r2, n2 ve P2 değişkenlerindeki değerleri dosyaya yaz,
11. $n3=3 \times 12$,
12. $P3=A \times r3 / (1 - (1+r3)^{-n3})$,
13. A, r3, n3, P3 ve $P3 \times n3$ değişkenlerindeki değerleri dosyaya yaz,
14. $n5=n5+5$,
15. $P5=A \times r5 / (1 - (1+r5)^{-n5})$,
16. A, r5, n5, P5 ve $P5 \times n5$ değişkenlerindeki değerleri dosyaya yaz,
17. $n10=n10+10$,
18. $P10=A \times r10 / (1 - (1+r10)^{-n10})$,
19. A, r10, n10, P10 ve $P10 \times n10$ değişkenlerindeki değerleri dosyaya yaz,
20. Son

Program 3-16 Dosyaya kredi faizlerini veri olarak yazan bir C++ programı.

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <math.h>
using namespace std;
```

```

int main() {
    float A=50000;
    float r1=0.0126, r2=0.0129, r3=0.0131, r5=0.0134, r10=0.0138;
    float P1, P2, P3, P5, P10;
    int n1=1*12, n2=2*12, n3=3*12, n5=5*12, n10=10*12;
    P1=A*r1/(1-pow((1+r1), -n1));
    P2=A*r2/(1-pow((1+r2), -n2));
    P3=A*r3/(1-pow((1+r3), -n3));
    P5=A*r5/(1-pow((1+r5), -n5));
    P10=A*r10/(1-pow((1+r10), -n10));
    ofstream dosyal; // dosyal yazma amaçlı tanımlanır
    dosyal.open ("veriler.txt", ios::out);
    dosyal << "Çekilen      Faiz      Toplam      Aylık      Toplam \n";
    dosyal << " Para      Oranı      Ay      Geri ödeme      Ödeme \n";
    dosyal << "-----\n";
    dosyal << fixed << setprecision(0);
    dosyal.width(4);
    dosyal.fill(' '); //belirtilen haneler boşlukla doldurulur
    dosyal << A << "      %" << setprecision(2) << r1*100 << "      ";
    dosyal << setprecision(0) << n1 << "      ";
    dosyal << setprecision(2) << P1 << "      " << P1*n1 << "\n";
    dosyal << fixed << setprecision(0);
    dosyal.width(4);
    dosyal.fill(' '); //belirtilen haneler boşlukla doldurulur
    dosyal << A << "      %" << setprecision(2) << r2*100 << "      ";
    dosyal << setprecision(0) << n2 << "      ";
    dosyal << setprecision(2) << P2 << "      " << P2*n2 << "\n";
    dosyal << fixed << setprecision(0);
    dosyal.width(4);
    dosyal.fill(' '); //belirtilen haneler boşlukla doldurulur
    dosyal << A << "      %" << setprecision(2) << r3*100 << "      ";
    dosyal << setprecision(0) << n3 << "      ";
    dosyal << setprecision(2) << P3 << "      " << P3*n3 << "\n";
    dosyal << fixed << setprecision(0);
    dosyal.width(4);
    dosyal.fill(' '); //belirtilen haneler boşlukla doldurulur
    dosyal << A << "      %" << setprecision(2) << r5*100 << "      ";
    dosyal << setprecision(0) << n5 << "      ";
    dosyal << setprecision(2) << P5 << "      " << P5*(n5+5) << "\n";
    dosyal << fixed << setprecision(0);
    dosyal.width(4);
}

```

```

dosyal.fill(' '); //belirtilen haneler boşlukla doldurulur
dosyal << A << "          %" << setprecision(2) << r10*100 << "          ";
dosyal << setprecision(0) << n10 << "          ";
dosyal << setprecision(2) << P10 << "          " << P10*(n10+10) << "\n";
dosyal.close();
return 0;}

```

Programın çalıştırılması sonucunda oluşturulan dosya içeriği aşağıdaki gibidir:

Çekilen Para	Faiz Oranı	Toplam Ay	Aylık Geri ödeme	Toplam Ödeme
50000	%1.26	12	4515.75	54188.97
50000	%1.29	24	2435.75	58458.02
50000	%1.31	36	1750.95	63034.11
50000	%1.34	60	1218.03	73081.75
50000	%1.38	120	855.10	102611.74

3.6 Karakter Katarları (strings)

Karakterler bellekte 1 byte yer işgal edecek şekilde saklanırlar. C++ da karakterler arka arkaya eklenerek (karakter katarı) kullanılırlar veya işlenirler. Bu katarlar char (karakter veri) tipindeki değişkenlerin diziler şeklinde tanımlanması ile oluşturulur. Dizinin her bir elemanına 1 karakter yerleştirilir. C-stilindeki karakter katarları veya string kütüphaneleri kullanılarak bellekte saklama ve diğer karakter işlemleri yapılabilir. Diziler kısmında bu konu daha ayrıntılı incelenektir.

Örnek olarak ekrana cout veya fprintf komutları ile yazdırılacak olan "Rakam giriniz" şeklindeki bir mesaj karakter katarıdır. Dizilerle kullanılan her stringin son karakteri veya dizinin son elemanı null karakter olmalıdır (\0).

Aşağıdaki Çizelge 3-5 de C++ da rezerve edilmiş (programcı tarafından kullanılmayan) fonksiyon isimleri verilmektedir. Kalın karakterlerle yazılmış deyimler standart C deki fonksiyonlardır.

Çizelge 3-5 C++ programlama dilinde rezerve edilmiş fonksiyon isimleri (Keywords).

asm	auto	bool	break	case
catch	char	class	const	const_cast
continue	default	delete	do double	
dynamic_cast	else enum	explicit	export	
extern	false	float	for	friend
goto	if	inline	int	long
mutable	namespace	new	operator	private
protected	public	register	reinterpret_cast	return
short	signed	sizeof	static	static_cast
struct	switch	template	this	throw
true	try	typedef	typeid	typename
union	unsigned	using	virtual	void
volatile	wchar_t	while		

3.7 SORULAR

1. Bir sınıftaki erkek ve kız öğrencilerinin boy ortalamalarını hesaplayan ve sonucu ekrana yazan bir C/C++ programını aşağıdaki algoritmayı kullanarak yazınız. Kız ve erkek öğrencilerin boy değerleri klavyeden bilgisayar belleğine aktarılmalıdır.

Algoritma 3-13 Erkek ve kız öğrencilerin boy ortalamaları.

1. Başla
2. float b, te=0, tk=0, eo, ko;
3. int ie=0, ik=0, i;
4. char c;
5. "Öğrencinin boyunu cm cinsinden giriniz:", b;
6. "Öğrencinin cinsiyetini giriniz:", c;
7. i=0;
8. Eğer (c=='E' || c=='e') ise i=1;
9. Eğer (c=='K' || c=='k') ise i=2;
10. switch(i){
 case 1 : {te=te+b; ie=ie+1; break;}
 case 2 : {tk=tk+b; ik=ik+1; break;}
 default : printf("işlem sonu\n"); Get 12; break;}
 } // swich case sonu
11. Get 5
12. eo=te/(float)ie; ko=tk/(float)ik;
13. Yaz " Erkeklerin boy ortalaması =", eo;
14. Yaz " Kızların boy ortalaması =", ko;
15. Son

2. 1 inch = 2.54 cm ve 1 cm = 10^{-2} m ise klavyeden girilen 8.5 inch³ lük hacim değerini m³ değerine çeviren bir C++ programını aşağıdaki algoritmayı kullanarak yazınız.

Algoritma 3-14 inch³ lük hacim değerinin m³ değerine dönüştürülmesi.

1. Başla
2. float a1, a2, a3, a4;
3. "Kübün inch biriminde hacim değerini giriniz:", a1;
4. a2=pow(a1, 1/3);
5. a3=2.54*a2*0.01;
6. a4=pow(a3, 3);
7. Yaz a1, " inch küp = ", a4, " metre küp";
8. Son

3. Klavyeden gün olarak girilen (kesirli bir) değeri (tam sayılar olarak) saat, dakika ve saniyeye çeviren C++ programını aşağıdaki algoritmayı kullanarak yazınız.

Algoritma 3-15 Günün saat dakika ve saniyeye dönüştürülmesi.

1. Başla
2. int saniye=3600, dakika=60, saat, s1, s2, d1, d2, kalan;
3. float G;
4. "Gün değişkenine değerini giriniz", G;
5. Eğer G<=0 ise Git 4
6. s1=G*24*saniye;
7. saat=s1/saniye;
8. kalan=s1-saat*saniye ;
9. s2=kalan;
10. d1=kalan/dakika;
11. kalan=s2-d1*dakika;
12. d2=kalan;
13. Yaz G, " gün =", saat, ", saat ", d1, " dakika ", d2, " saniye";
14. Son.

4. Aşağıdaki çizelgedeki gibi bir dosyadan okunan verileri kullanarak değişik maddelerden yapılmış silindirlere yoğunluğunu hesaplayan programı aşağıdaki algoritmayı kullanarak yazınız.

Madde	Kütle (kg)	Yarıçap (cm)	Uzunluk (cm)
Alüminyum	51.5	2.53	3.75
Bakır	56.3	1.23	5.06
Pirinç	94.4	1.54	5.69
Teneke	69.1	1.75	3.74
Demir	216.1	1.89	9.77

Algoritma 3-16 Dosyadan okunan bilgilerden maddelerin yoğunluğunun hesaplanması.

1. Başla
2. int main()
3. char DosyaAdi[20]; // dosya adını saklama değişkeni
4. char satir[80]; // dosyadaki her satırı okuma (80 sütun)
5. ifstream dosyal; // dosyal tanımlanır
6. Yaz "Dosya adını giriniz:";
7. cin.getline (DosyaAdi, sizeof (DosyaAdi));
8. dosyal.open (DosyaAdi, ios::in); // dosya okuma amaçlı açılır
9. Eğer (dosyal.eof()) ise Git 13
10. dosyal.getline(satir, sizeof (satir));
11. cout << satir << endl;
12. Git 9
13. dosyal.close (); // dosyal kapatılır
14. return 0; // işletim sistemine hata yok bilgisi yollanır
15. Son

5. Dörtgen bir prizmaya ait kenar uzunlukları klavyeden girildikten (yani a, b ve c değişkenlerine aktarıldıktan) sonra bu dörtgen prizmanın köşegen uzunluğunu ($R = \sqrt{a^2 + b^2 + c^2}$) denklemine göre hesaplayan C/C++ programını aşağıdaki algoritmayı kullanarak yazınız.

Algoritma 3-17 Dörtgen prizmanın köşegen uzunluğu.

1. Başla
2. float a, b, c, R;
3. "Dörtgen prizmanın kenar değerlerini giriniz", a, b, c;
4. $R = \sqrt{a^2 + b^2 + c^2}$;
5. Yaz " Dörtgen prizmanın köşegen uzunluğu = ", R;
6. Son.

C/C++ Ders Notları-A.Ü. Fizik Mühendisliği Bölümü