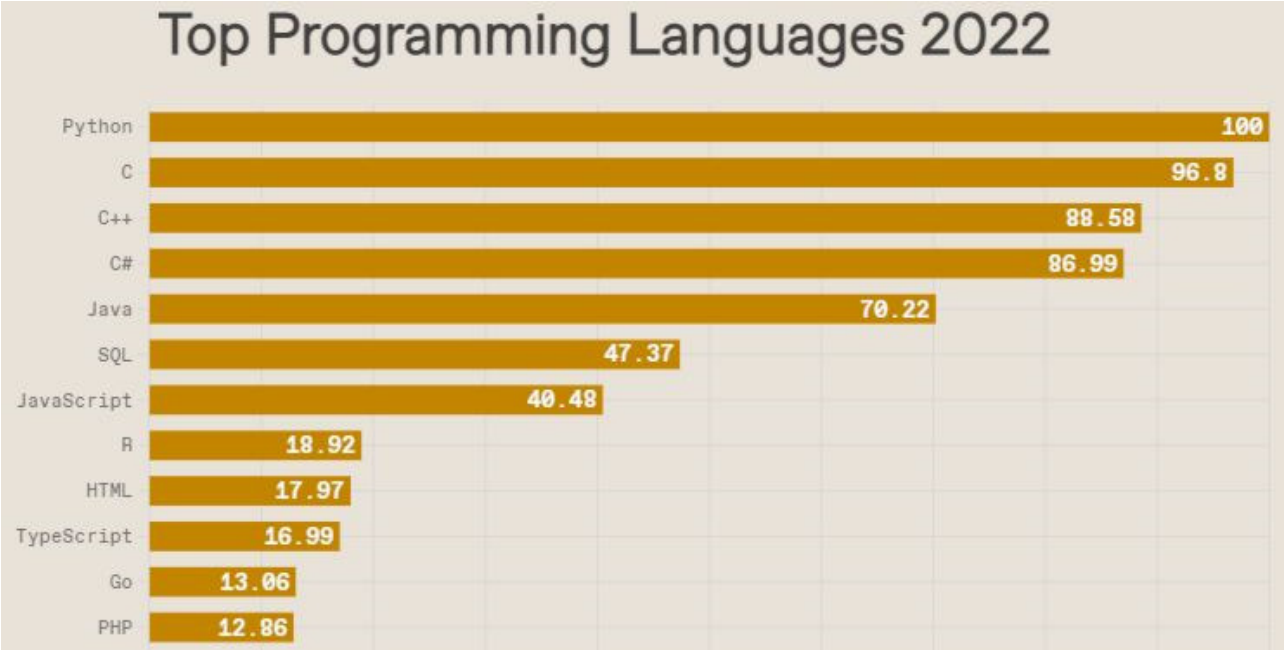
Yazılım Mühendisliğine (YM) Giriş

Ders Notu V

2023 GÜZ

IEEE
Advancing Technology for Humanity



IEEE Spectrum Ranks Languages

IEEE: The Institute of Electrical and Electronics Engineers

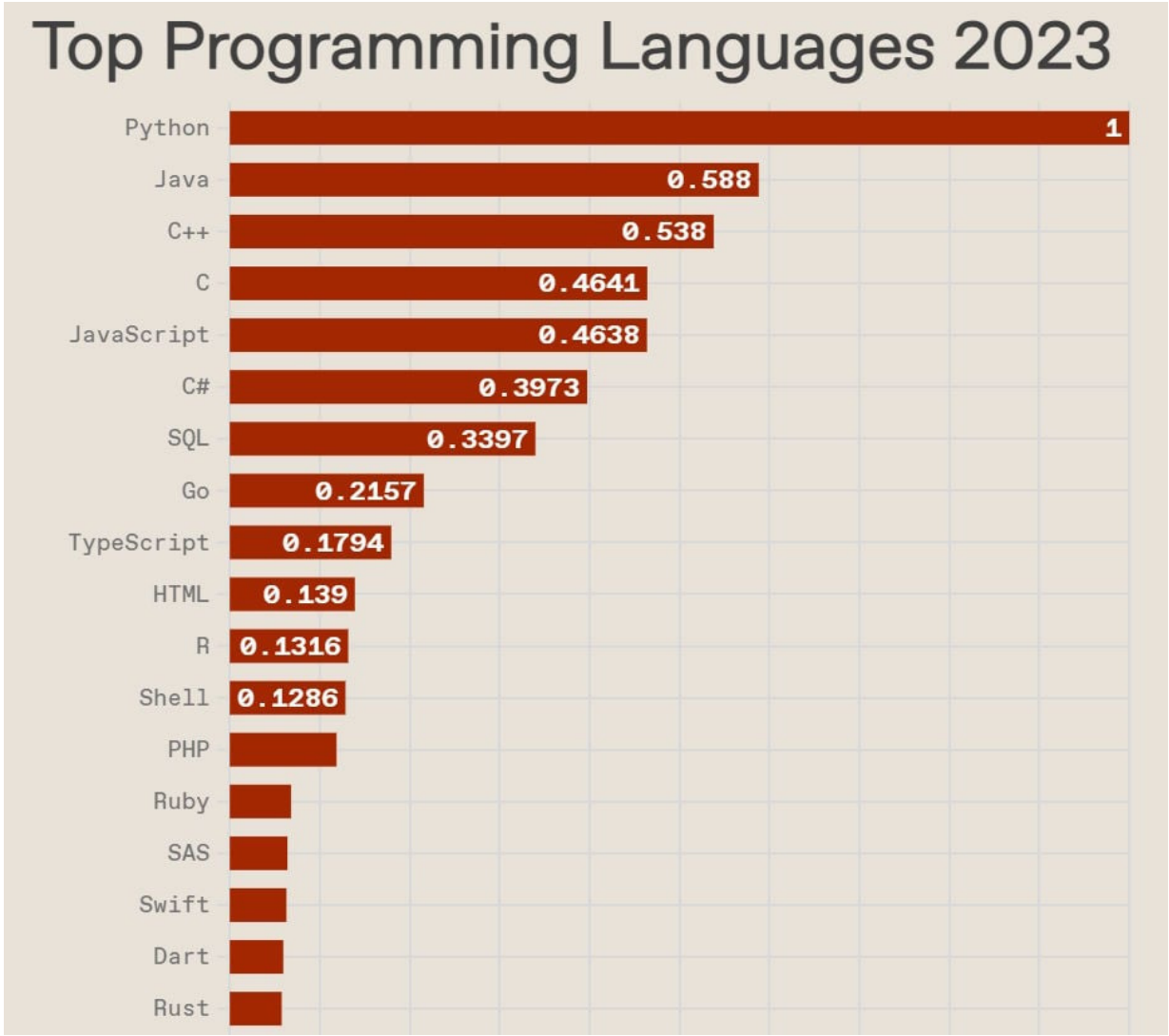
```
print('Hello, World!')
```

```
class HelloWorld
{
  public static void main(String[] args)
  {   System.out.println("Hello,
World!");   }
}
```

```
<script>
console.log('Hello, World!');
</script>
```

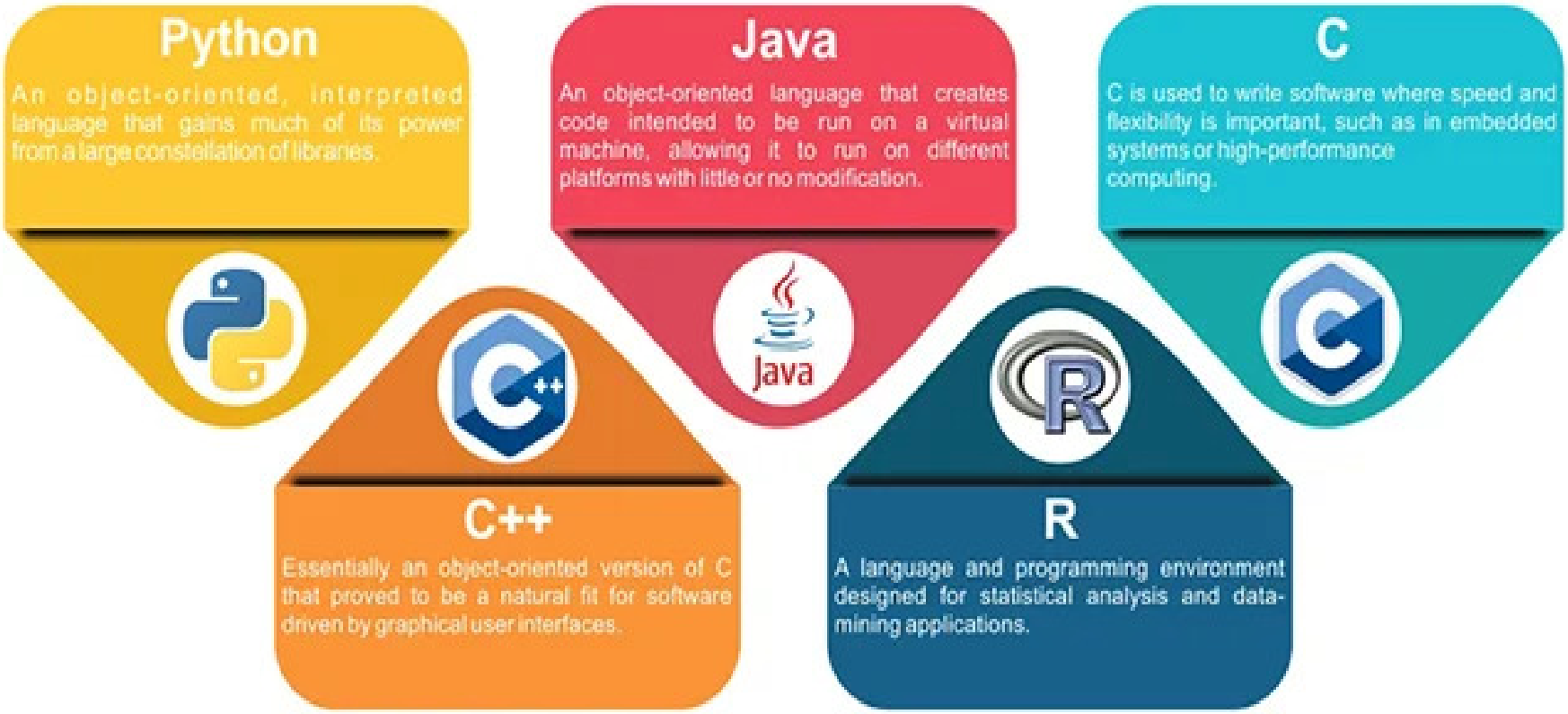
```
#include <iostream>
using namespace std;

int main()

{
  cout << "Hello, World";
  return 0;
}
```

IEEE
Advancing Technology for Humanity



The November 2023 issue of *IEEE Spectrum*

IEEE Top Programming Languages in 2019

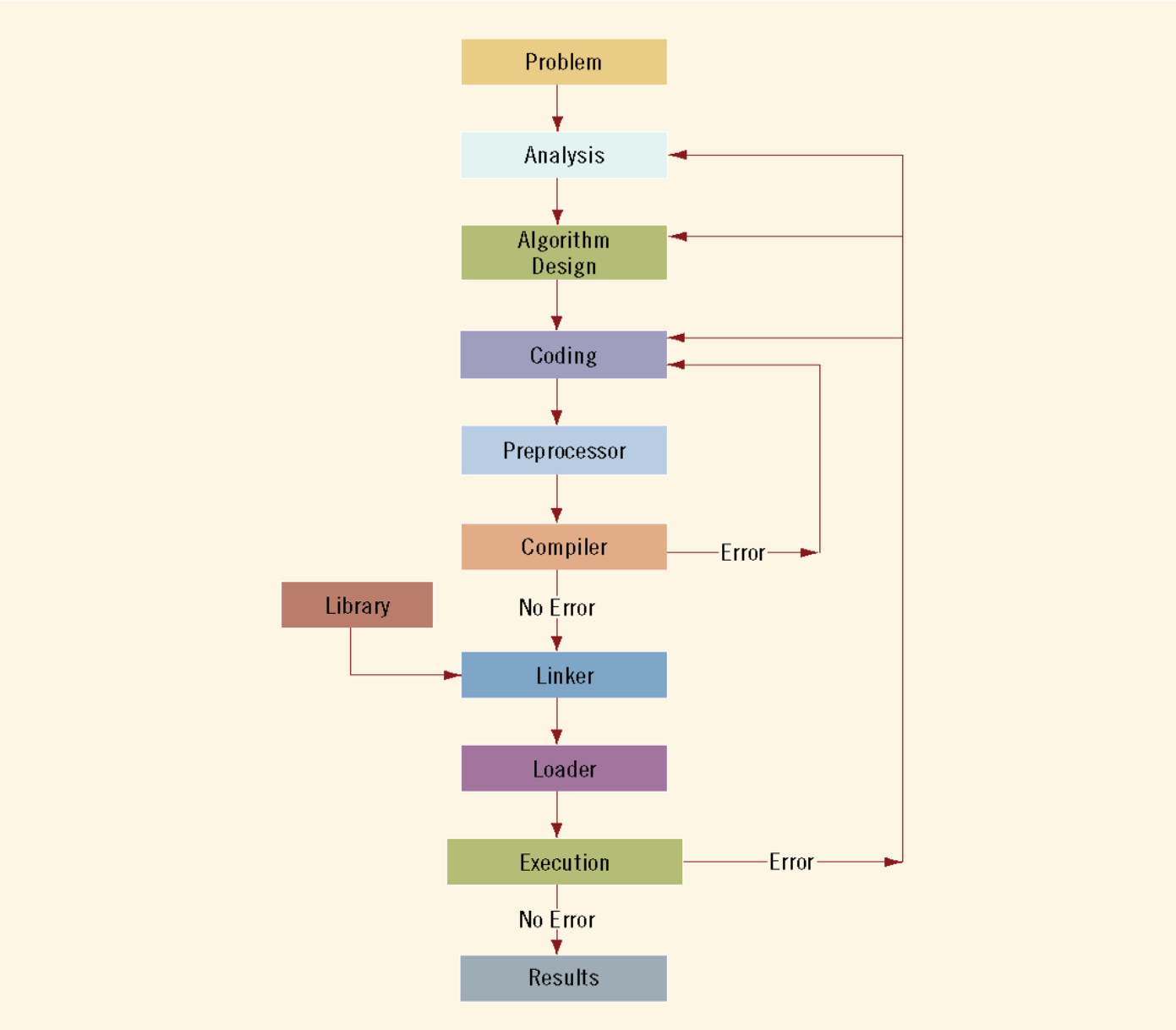


Problem Çözme

- Programlama bir problem çözme prosesidir.
- Problem çözme teknikleri
  - Problem analiz edilir
  - Problemin gereksinimleri belirlenir
  - Problemin çözümü için tasarım adımına (algoritmaya) geçilir.
- Algoritma :
  - Adım adım problem çözme prosesidir.
  - Çözüme sonlu bir zaman diliminde ulaşılır.

Problem Çözme Prosesi

- Adım 1 – Problem analiz edilir.
  - Problem analiz edilir
  - Problemin gereksinimleri belirlenir
  - Problemin çözümü için tasarım adımına (algoritmaya) geçilir.
- Adım 2 – Algoritma çalıştırılır (implementasyon)
  - Algoritma koda dönüştürülür.
  - Algoritmanın çalıştığı kontrol edilir (verify)
- Adım 3 – Devamlılığın sağlanması (Maintenance)
  - Problemin uygulama alanı değiştiği taktirde programın kullanılabilmesi ve değiştirilebilmesi



Problem analizi ve kodunun çalıştırılması aşamaları

Problemin Analizi

- Baştan sona kadar problemin tümüyle anlaşılması
- Problem gereksinimlerinin açık olarak belirlenmesi
  - Program kullanıcı ile etkileşim gerektiriyor mu?
  - Program data üzerinde manipülasyon gerçekleştirebiliyor mu?
  - Elde edilen çıktı nedir?
- Problem karmaşık olduğunda alt problemlere bölünür.
  - Her bir alt problem yukarıda açıklandığı şekilde analiz edilir.

Algoritmanın Tasarımı

- Eğer problem alt problemlere bölündü ise
  - Her alt probleme ait algoritma tasarlanır.
- Algoritmanın doğruluğunun kontrol edilmesi
  - Bu süreç örnek veriler (data) kullanılarak gerçekleştirilir.
  - Bazı matematiksel analizlerin yapılması gerekebilir.

Kodun Yazılması

- Algoritma yazıldığında ve doğruluğu sağlandığında (verified)
  - Algoritmaya denk yüksek düzey bir dilde (örneğin C++) kodun yazımı gerçekleştirilir.
  - Programın yazılışı bir metin (text) editörü üzerinde gerçekleştirilir.

Derleme ve Bağlama
Compiling and Linking

- Kod derleyicide çalıştırılır.
  - Derleme sonunda hata ile karşılaşılırsa (errors)
    - Koda geri dönülerek hatalar kaldırılır.
    - Kod derleyicide yeniden çalıştırılır, yani yeniden derlenir.
  - Herhangi bir sözdizimi, yani yazım hatası kalmadığında
    - ✓ Derleyici kendisine denk makine kodunu oluşturur (generates).
- Linker ise, makine kodunu sistem kaynakları (system resources) ile bağlar.

Yükleyici ve Çalıştırma
(Loader and Executing)

- Program derlendiğinde ve sistem kaynakları ile birleştirildiğinde, loader (yükleyici) programı çalıştırmak üzere ana belleğe gelir.
- Son adım programın çalıştırılması adımıdır.
- Kodun derleme aşaması başarı ile tamamlandığı için programın dilin kurallarına uygun olduğu garanti edilmiştir.
  - Fakat programın doğru çalışacağının garantisi yoktur

???

Yapısal Programlama

- Yapısal Tasarım (Structured design)
  - Problem daha küçük alt problemlere bölünür
- Yapısal Programlama (Structured programming)
  - Yapısal tasarımın implementasyonu gerçekleştirilir
- Yapısal tasarım farklı şekillerde de adlandırılır:
  - Yukarıdan aşağıya doğru tasarım (Top-down design)
  - Adım adım yenileme (Stepwise refinement)
  - Modüler programlama (Modular programming)