1.8

using System;

public class Time

{

private int hour;

private int minute;

private int second;

public Time(int hour, int minute, int second)

{

SetTime(hour, minute, second);

}

public void SetHour(int hour)

{

if (hour >= 0 && hour < 24)

{

this.hour = hour;

}

}

public int GetHour()

{

return hour;

}

public void SetMinute(int minute)

{

if (minute >= 0 && minute < 60)

{

this.minute = minute;

}

}

public int GetMinute()

{

return minute;

}

public void SetSecond(int second)

{

if (second >= 0 && second < 60)

{

this.second = second;

}

}

public int GetSecond()

{

return second;

}

public void SetTime(int hour, int minute, int second)

{

SetHour(hour);

SetMinute(minute);

SetSecond(second);

}

public Time NextSecond()

{

second++;

if (second == 60)

{

second = 0;

minute++;

if (minute == 60)

{

minute = 0;

hour++;

if (hour == 24)

{

hour = 0;

}

}

}

return this;

}

public Time PreviousSecond()

{

second--;

if (second < 0)

{

second = 59;

minute--;

if (minute < 0)

{

minute = 59;

hour--;

if (hour < 0)

{

hour = 23;

}

}

}

return this;

}

public override string ToString()

{

return string.Format("{0:D2}:{1:D2}:{2:D2}", hour, minute, second);

}

}

public class TestMain

{

public static void Main(string[] args)

{

// Test constructors and ToString()

Time t1 = new Time(1, 2, 3);

Console.WriteLine(t1); // ToString()

// Test Setters and Getters

t1.SetHour(4);

t1.SetMinute(5);

t1.SetSecond(6);

Console.WriteLine(t1); // ToString()

Console.WriteLine("Hour: " + t1.GetHour());

Console.WriteLine("Minute: " + t1.GetMinute());

Console.WriteLine("Second: " + t1.GetSecond());

// Test SetTime()

t1.SetTime(23, 59, 58);

Console.WriteLine(t1); // ToString()

// Test NextSecond();

Console.WriteLine(t1.NextSecond());

Console.WriteLine(t1.NextSecond().NextSecond());

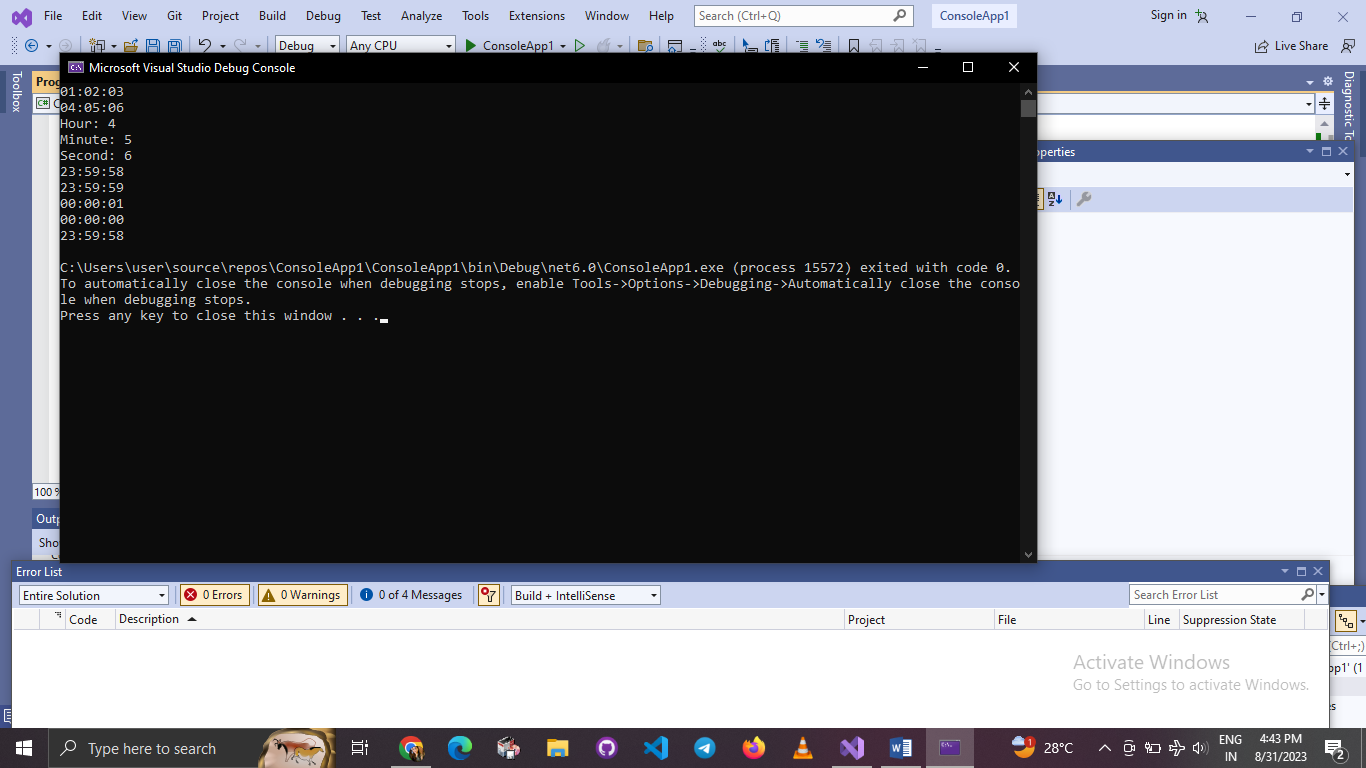
// Test PreviousSecond()

Console.WriteLine(t1.PreviousSecond());

Console.WriteLine(t1.PreviousSecond().PreviousSecond());

}

}



1.9

using System;

public class Ball

{

private float x;

private float y;

private int radius;

private float xDelta;

private float yDelta;

public Ball(float x, float y, int radius, float xDelta, float yDelta)

{

this.x = x;

this.y = y;

this.radius = radius;

this.xDelta = xDelta;

this.yDelta = yDelta;

}

public void SetX(float x)

{

this.x = x;

}

public float GetX()

{

return x;

}

public void SetY(float y)

{

this.y = y;

}

public float GetY()

{

return y;

}

public void SetRadius(int radius)

{

this.radius = radius;

}

public int GetRadius()

{

return radius;

}

public void SetXDelta(float xDelta)

{

this.xDelta = xDelta;

}

public float GetXDelta()

{

return xDelta;

}

public void SetYDelta(float yDelta)

{

this.yDelta = yDelta;

}

public float GetYDelta()

{

return yDelta;

}

public void ReflectHorizontal()

{

xDelta = -xDelta;

}

public void ReflectVertical()

{

yDelta = -yDelta;

}

public void Move()

{

x += xDelta;

y += yDelta;

}

public override string ToString()

{

return $"Ball[({x},{y}),speed=({xDelta},{yDelta})]";

}

}

public class TestMain

{

public static void Main(string[] args)

{

// Test constructor and ToString()

Ball ball = new Ball(1.1f, 2.2f, 10, 3.3f, 4.4f);

Console.WriteLine(ball); // ToString()

// Test Setters and Getters

ball.SetX(80.0f);

ball.SetY(35.0f);

ball.SetRadius(5);

ball.SetXDelta(4.0f);

ball.SetYDelta(6.0f);

Console.WriteLine(ball); // ToString()

Console.WriteLine("x is: " + ball.GetX());

Console.WriteLine("y is: " + ball.GetY());

Console.WriteLine("radius is: " + ball.GetRadius());

Console.WriteLine("xDelta is: " + ball.GetXDelta());

Console.WriteLine("yDelta is: " + ball.GetYDelta());

// Bounce the ball within the boundary

float xMin = 0.0f;

float xMax = 100.0f;

float yMin = 0.0f;

float yMax = 50.0f;

for (int i = 0; i < 15; i++)

{

ball.Move();

Console.WriteLine(ball);

float xNew = ball.GetX();

float yNew = ball.GetY();

int radius = ball.GetRadius();

// Check boundary value to bounce back

if ((xNew + radius) > xMax || (xNew - radius) < xMin)

{

ball.ReflectHorizontal();

}

if ((yNew + radius) > yMax || (yNew - radius) < yMin)

{

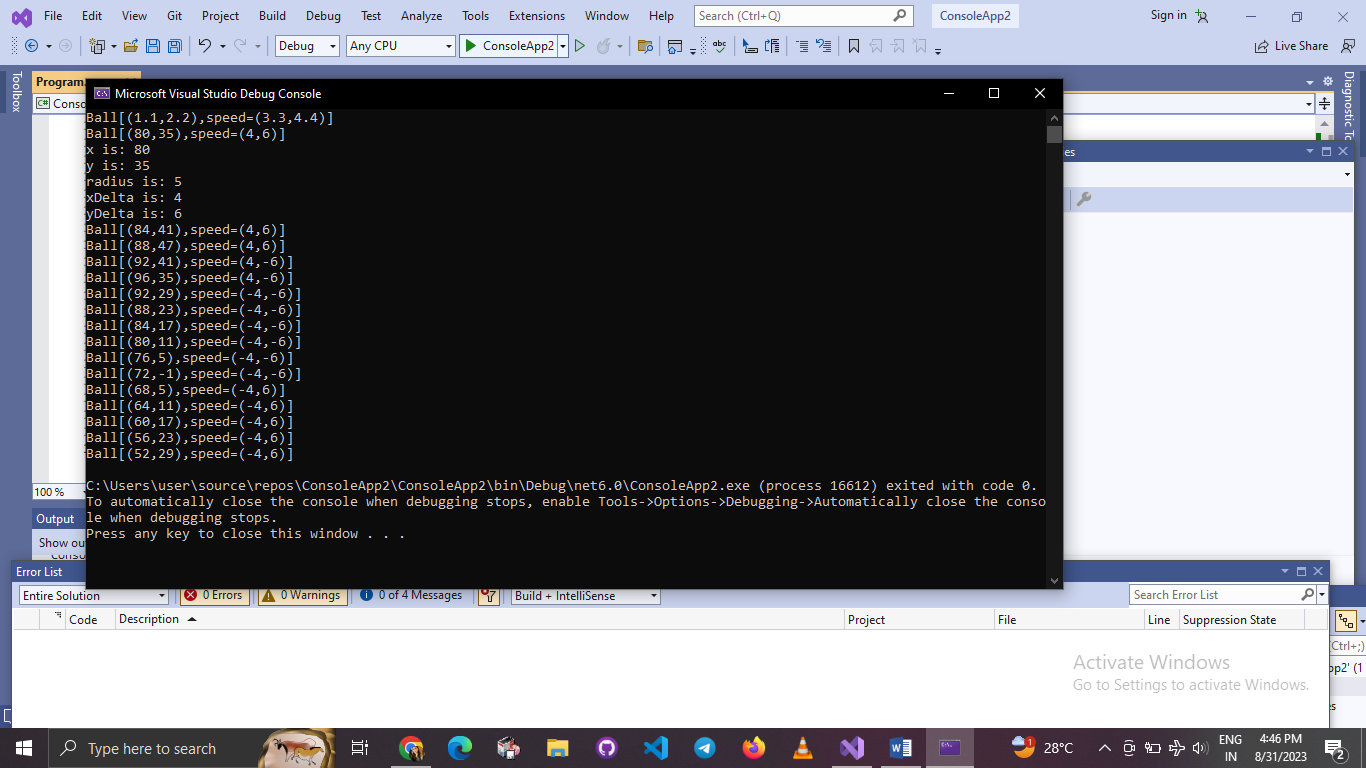
ball.ReflectVertical();

}

}

}

}



2.3

using System;

class Author

{

private string name;

private string email;

public Author(string name, string email)

{

this.name = name;

this.email = email;

}

public string GetName()

{

return name;

}

public string GetEmail()

{

return email;

}

public void SetEmail(string email)

{

this.email = email;

}

public override string ToString()

{

return $"Author[name={name},email={email}]";

}

}

class Book

{

private string isbn;

private string name;

private Author author;

private double price;

private int qty;

public Book(string isbn, string name, Author author, double price, int qty)

{

this.isbn = isbn;

this.name = name;

this.author = author;

this.price = price;

this.qty = qty;

}

public string GetIsbn()

{

return isbn;

}

public string GetName()

{

return name;

}

public double GetPrice()

{

return price;

}

public void SetPrice(double price)

{

this.price = price;

}

public int GetQty()

{

return qty;

}

public void SetQty(int qty)

{

this.qty = qty;

}

public Author GetAuthor()

{

return author;

}

public string GetAuthorName()

{

return author.GetName();

}

public override string ToString()

{

return $"Book[isbn={isbn},name={name},{author},price={price},qty={qty}]";

}

}

class TestMain

{

static void Main(string[] args)

{

// Test Author class

Author a1 = new Author("Tan Ah Teck", "ahteck@nowhere.com");

Console.WriteLine(a1);

a1.SetEmail("ahteck@somewhere.com");

Console.WriteLine(a1);

Console.WriteLine("name is: " + a1.GetName());

Console.WriteLine("email is: " + a1.GetEmail());

// Test Book class

Book b1 = new Book("12345", "Java for dummies", a1, 8.8, 88);

Console.WriteLine(b1);

b1.SetPrice(9.9);

b1.SetQty(99);

Console.WriteLine(b1);

Console.WriteLine("isbn is: " + b1.GetIsbn());

Console.WriteLine("name is: " + b1.GetName());

Console.WriteLine("price is: " + b1.GetPrice());

Console.WriteLine("qty is: " + b1.GetQty());

Console.WriteLine("author is: " + b1.GetAuthor());

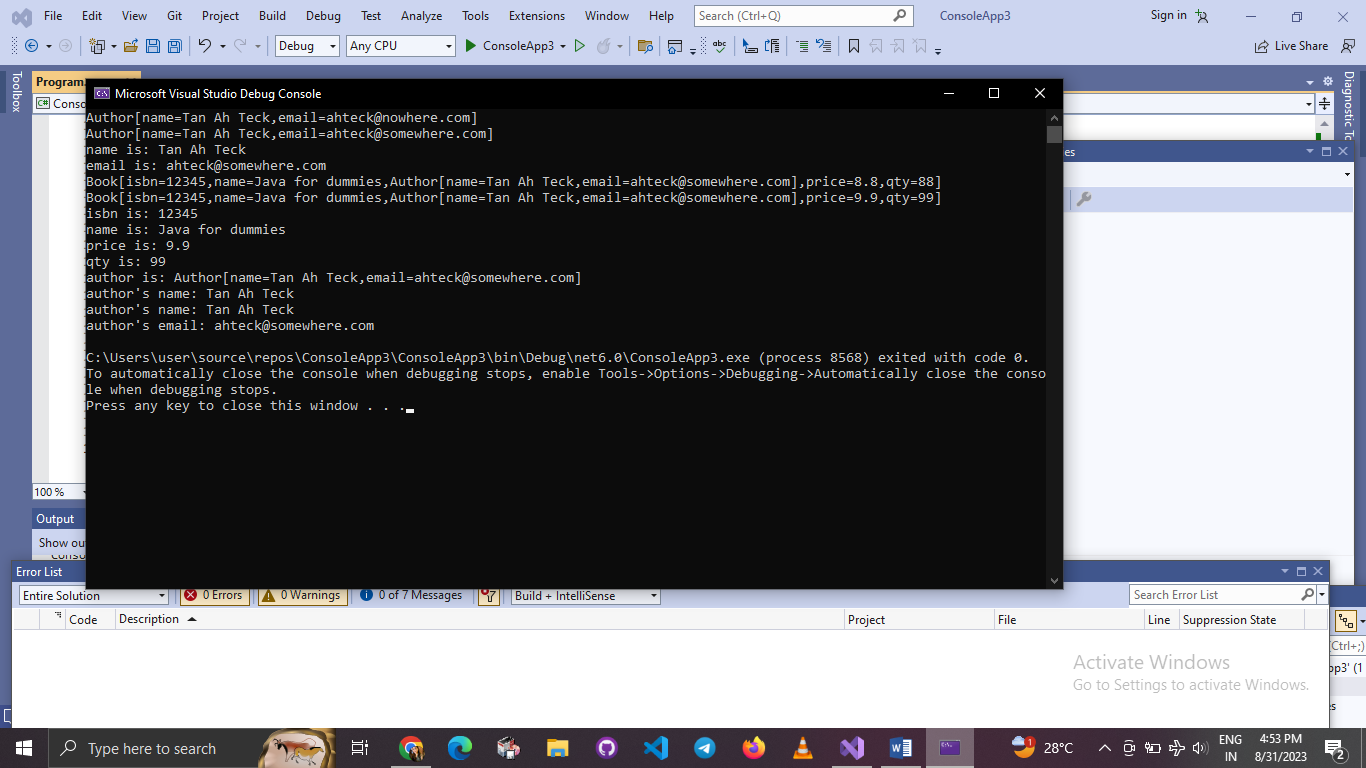
Console.WriteLine("author's name: " + b1.GetAuthorName());

Console.WriteLine("author's name: " + b1.GetAuthor().GetName());

Console.WriteLine("author's email: " + b1.GetAuthor().GetEmail());

}

}



2.5

using System;

class Customer

{

private int id;

private string name;

private int discount;

public Customer(int id, string name, int discount)

{

this.id = id;

this.name = name;

this.discount = discount;

}

public int GetId()

{

return id;

}

public string GetName()

{

return name;

}

public int GetDiscount()

{

return discount;

}

public void SetDiscount(int discount)

{

this.discount = discount;

}

public override string ToString()

{

return $"{name}({id})({discount}%)";

}

}

class Invoice

{

private int id;

private Customer customer;

private double amount;

public Invoice(int id, Customer customer, double amount)

{

this.id = id;

this.customer = customer;

this.amount = amount;

}

public int GetId()

{

return id;

}

public Customer GetCustomer()

{

return customer;

}

public double GetAmount()

{

return amount;

}

public void SetAmount(double amount)

{

this.amount = amount;

}

public int GetCustomerId()

{

return customer.GetId();

}

public string GetCustomerName()

{

return customer.GetName();

}

public int GetCustomerDiscount()

{

return customer.GetDiscount();

}

public double GetAmountAfterDiscount()

{

return amount - (amount \* customer.GetDiscount() / 100.0);

}

public override string ToString()

{

return $"Invoice[id={id},customer={customer},amount={amount}]";

}

}

class TestMain

{

static void Main(string[] args)

{

// Test Customer class

Customer c1 = new Customer(88, "Tan Ah Teck", 10);

Console.WriteLine(c1);

c1.SetDiscount(8);

Console.WriteLine(c1);

Console.WriteLine("id is: " + c1.GetId());

Console.WriteLine("name is: " + c1.GetName());

Console.WriteLine("discount is: " + c1.GetDiscount());

// Test Invoice class

Invoice inv1 = new Invoice(101, c1, 888.8);

Console.WriteLine(inv1);

inv1.SetAmount(999.9);

Console.WriteLine(inv1);

Console.WriteLine("id is: " + inv1.GetId());

Console.WriteLine("customer is: " + inv1.GetCustomer());

Console.WriteLine("amount is: " + inv1.GetAmount());

Console.WriteLine("customer's id is: " + inv1.GetCustomerId());

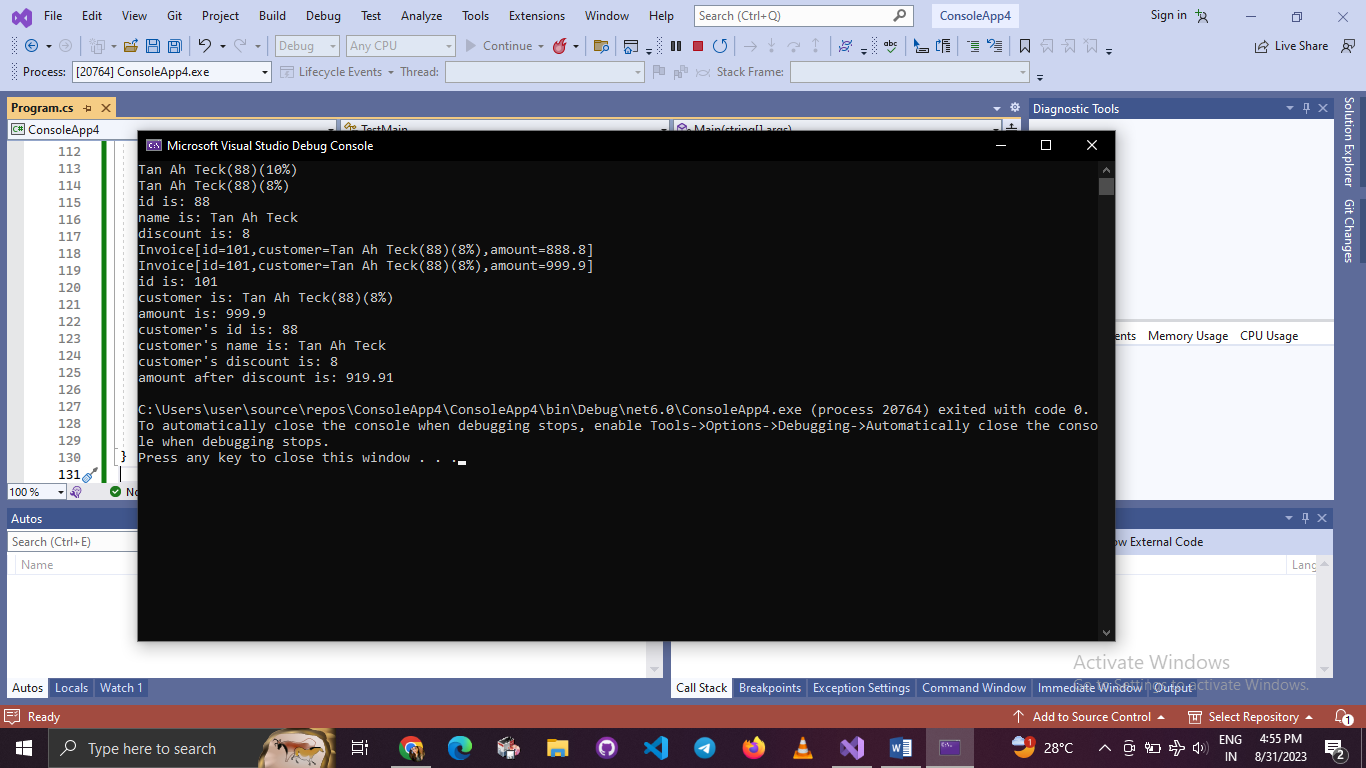
Console.WriteLine("customer's name is: " + inv1.GetCustomerName());

Console.WriteLine("customer's discount is: " + inv1.GetCustomerDiscount());

Console.WriteLine("amount after discount is: " + inv1.GetAmountAfterDiscount().ToString("0.00"));

}

}



2.1

using System;

class Author

{

private string name;

private string email;

private char gender;

public Author(string name, string email, char gender)

{

this.name = name;

this.email = email;

this.gender = gender;

}

public string GetName()

{

return name;

}

public string GetEmail()

{

return email;

}

public char GetGender()

{

return gender;

}

public override string ToString()

{

return $"Author[name={name},email={email},gender={gender}]";

}

}

class Book

{

private string name;

private Author author;

private double price;

private int qty;

public Book(string name, Author author, double price)

{

this.name = name;

this.author = author;

this.price = price;

this.qty = 0; // Default qty value

}

public Book(string name, Author author, double price, int qty)

{

this.name = name;

this.author = author;

this.price = price;

this.qty = qty;

}

public string GetName()

{

return name;

}

public Author GetAuthor()

{

return author;

}

public double GetPrice()

{

return price;

}

public void SetPrice(double price)

{

this.price = price;

}

public int GetQty()

{

return qty;

}

public void SetQty(int qty)

{

this.qty = qty;

}

public override string ToString()

{

return $"Book[name={name},{author},price={price},qty={qty}]";

}

public string GetAuthorName()

{

return author.GetName();

}

public string GetAuthorEmail()

{

return author.GetEmail();

}

public char GetAuthorGender()

{

return author.GetGender();

}

}

class TestBook

{

static void Main(string[] args)

{

// Construct an author instance

Author ahTeck = new Author("Tan Ah Teck", "ahteck@nowhere.com", 'm');

Console.WriteLine(ahTeck); // Author's toString()

Book dummyBook = new Book("Java for dummy", ahTeck, 19.95, 99); // Test Book's Constructor

Console.WriteLine(dummyBook); // Test Book's toString()

// Test Getters and Setters

dummyBook.SetPrice(29.95);

dummyBook.SetQty(28);

Console.WriteLine("name is: " + dummyBook.GetName());

Console.WriteLine("price is: " + dummyBook.GetPrice());

Console.WriteLine("qty is: " + dummyBook.GetQty());

Console.WriteLine("Author is: " + dummyBook.GetAuthor()); // Author's toString()

Console.WriteLine("Author's name is: " + dummyBook.GetAuthorName());

Console.WriteLine("Author's email is: " + dummyBook.GetAuthorEmail());

// Use an anonymous instance of Author to construct a Book instance

Book anotherBook = new Book("more Java",

new Author("Paul Tan", "paul@somewhere.com", 'm'), 29.95);

Console.WriteLine(anotherBook); // toString()

// Print name and email of author from a Book instance

Console.WriteLine("Author's name from Book: " + dummyBook.GetAuthorName());

Console.WriteLine("Author's email from Book: " + dummyBook.GetAuthorEmail());

}

}

