



Universidad
del Valle de México

LAUREATE INTERNATIONAL UNIVERSITIES®

Actividad 5

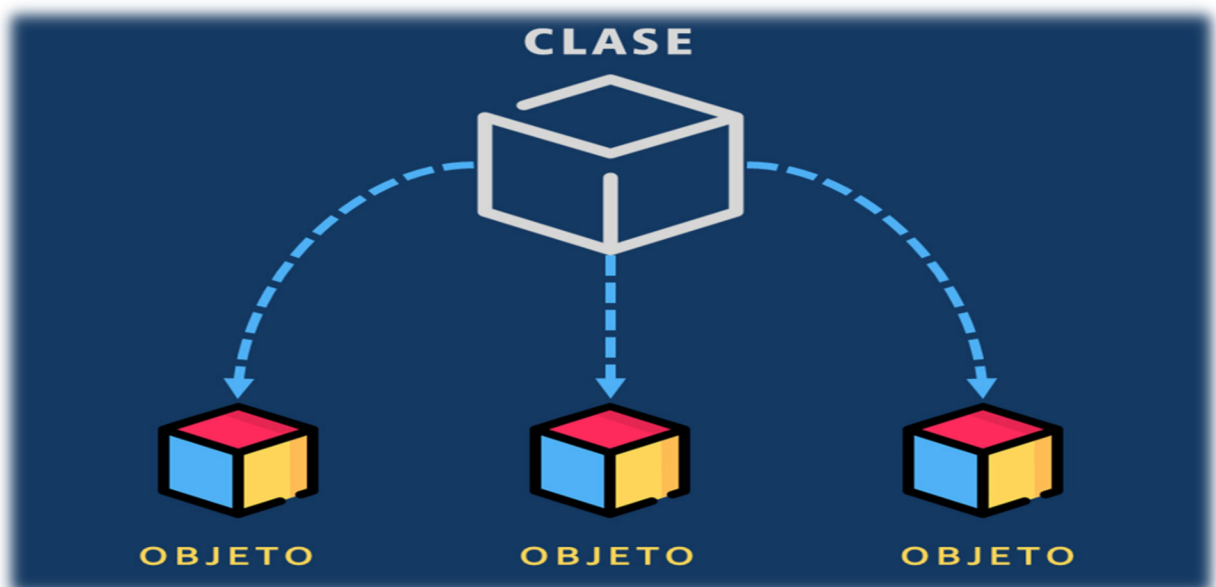
Ejercicios

Programación orientada a objetos

Docente. Ing. Saúl Santiago Rivera

José Emiliano Jauregui Guzmán

Por siempre responsable de lo que se ha cultivado



ACTIVIDAD V: EJERCICIOS

1. Con base en el material consultado en la unidad resuelve los siguientes ejercicios que se plantean a continuación acerca de los siguientes temas:

- Lenguaje C++
- Estructura y sobrecarga de métodos

Ejercicio 1.

- a) Escribe un programa que calcule el área de distintas figuras geométricas (triángulo, cuadrado, círculo y trapecio) para ello **sobrecarga** la función o método “área” con el número y tipo de argumentos necesarios para cada figura.

```
main.cpp
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5
6  // Área del triángulo
7  double area(double base, double altura) {
8      return (base * altura) / 2;
9  }
10
11 // Área del cuadrado
12 double area(double lado) {
13     return lado * lado;
14 }
15
16 // Área de un círculo
17 double area(double radio, bool esCirculo) {
18     return M_PI * radio * radio;
19 }
20
21 // Área de un trapecio
22 double area(double baseMayor, double baseMenor, double altura) {
23     return (baseMayor + baseMenor) * altura / 2;
24 }
```

- b) En la función principal *main ()* integra un menú para que el usuario elija qué área quiere calcular.

```
main.cpp
25
26 int main() {
27     int opcion;
28     //Usamos el bucle do while para que siga mostrando el menú
29     do {
30         cout << "Seleccione la figura para calcular el área: " << endl;
31         cout << "1 Área Triángulo" << endl;
32         cout << "2 Área Cuadrado" << endl;
33         cout << "3 Área Círculo" << endl;
34         cout << "4 Área Trapecio" << endl;
35         cout << "5 Salir" << endl;
36         cout << "Opción: ";
37         cin >> opcion;
38
39
40         double base, altura, radio, lado, baseMayor, baseMenor;
41     }
```



```

Seleccione la figura para calcular el área:
1 Área Triángulo
2 Área Cuadrado
3 Área Círculo
4 Área Trapecio
5 Salir
Opción: 1

Ingrese la base del triángulo: 2
Ingrese la altura del triángulo: 2
El área del triángulo es: 2

Seleccione la figura para calcular el área:
1 Área Triángulo
2 Área Cuadrado
3 Área Círculo
4 Área Trapecio
5 Salir
Opción: 2

Ingrese el lado del cuadrado: 2
El área del cuadrado es: 4

Seleccione la figura para calcular el área:
1 Área Triángulo
2 Área Cuadrado
3 Área Círculo
4 Área Trapecio
5 Salir
Opción: 3
Ingrese el radio del círculo: 4
El área del círculo es: 50.2655

Seleccione la figura para calcular el área:
1 Área Triángulo
2 Área Cuadrado
3 Área Círculo
4 Área Trapecio
5 Salir
Opción: 4
Ingrese la base mayor del trapecio:
4
Ingrese la base menor del trapecio: 6
Ingrese la altura del trapecio: 5
El área del trapecio es: 25

```

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
// Área de un triángulo
```

```
double area(double base, double altura) {
    return (base * altura) / 2;
}
```

```
// Área de un cuadrado
```

```
double area(double lado) {
    return lado * lado;
}
```

```
// Área de un círculo
```



```
double area(double radio, bool esCirculo) {
    return M_PI * radio * radio;
}

// Área de un trapecio
double area(double baseMayor, double baseMenor, double altura) {
    return (baseMayor + baseMenor) * altura / 2;
}

int main() {
    int opcion;

    do {
        // Menú interactivo
        cout << "\nSelecione la figura para calcular el área: " << endl;
        cout << "1 Área Triángulo" << endl;
        cout << "2 Área Cuadrado" << endl;
        cout << "3 Área Círculo" << endl;
        cout << "4 Área Trapecio" << endl;
        cout << "5 Área Salir" << endl;
        cout << "Opción: ";
        cin >> opcion;

        // Variables locales para las dimensiones
        double base, altura, radio, lado, baseMayor, baseMenor;

        switch (opcion) {
            case 1:
                cout << "\nIngrese la base del triángulo: ";
                cin >> base;
                cout << "Ingrese la altura del triángulo: ";
                cin >> altura;
                cout << "El área del triángulo es: " << area(base, altura) << endl;
                break;
        }
    } while (opcion != 5);
}
```



case 2:

```
cout << "\nIngrese el lado del cuadrado: ";
cin >> lado;
cout << "El área del cuadrado es: " << area(lado) << endl;
break;
```

case 3:

```
cout << "Ingrese el radio del círculo: ";
cin >> radio;
cout << "El área del círculo es: " << area(radio, true) << endl;
break;
```

case 4:

```
cout << "Ingrese la base mayor del trapecio: ";
cin >> baseMayor;
cout << "Ingrese la base menor del trapecio: ";
cin >> baseMenor;
cout << "Ingrese la altura del trapecio: ";
cin >> altura;
cout << "El área del trapecio es: " << area(baseMayor, baseMenor, altura) << endl;
break;
```

case 5:

```
cout << "Saliendo del programa..." << endl;
break;
```

default:

```
cout << "Opción no válida. Intente de nuevo." << endl;
```

```
}
```

```
} while (opcion != 5);
```

```
return 0;
```

```
}
```



Ejercicio 2.

a) Define la Clase “persona” con atributos, nombre, edad y sexo.

```
main.cpp
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  class Persona {
6  public:
7      // Atributos
8      string nombre;
9      int edad;
10     char sexo;
11
12     Persona(string nombre, int edad, char sexo) : nombre(nombre), edad(edad), sexo(sexo) {}
13 }
```

b) Programa una función imprimir que reciba como parámetro un objeto de la Clase “persona” e imprima los valores, que previamente haya solicitado desde el teclado. Utilizar void imprimir (Persona persona_imp)

```
main.cpp
12  Persona(string nombre, int edad, char sexo) : nombre(nombre), edad(edad), sexo(sexo) {}
13
14  // Método para solicitar y almacenar los valores de la persona
15  void solicitarDatos() {
16      cout << "Ingrese el nombre: ";
17      //getline porque el nombre puede contener espacios.
18      getline(cin, nombre);
19      cout << "Ingrese la edad: ";
20      cin >> edad;
21      cout << "Ingrese el sexo (Masculino/Femenino): ";
22      cin >> sexo;
23      cin.ignore();
24  }
25
26
27  void imprimir(Persona persona_imp) {
28      cout << "\nDatos ingresados:" << endl;
29      cout << "Nombre: " << persona_imp.nombre << endl;
30      cout << "Edad: " << persona_imp.edad << endl;
31      cout << "Sexo: " << persona_imp.sexo << endl;
32  }
33
34  int main() {
35      Persona persona1("", 0, ' ');
36      persona1.solicitarDatos();
37
38      Persona persona2("", 0, ' ');
39      persona2.solicitarDatos();
40
41      imprimir(persona1);
42      imprimir(persona2);
43
44      return 0;
45  }
46 }
```

Ingrese el nombre: Emiliano
Ingrese la edad: 23
Ingrese el sexo (Masculino/Femenino): M
Ingrese el nombre: Valeria
Ingrese la edad: 22
Ingrese el sexo (Masculino/Femenino): F

Datos ingresados:
Nombre: Emiliano
Edad: 23
Sexo: M

Datos ingresados:
Nombre: Valeria
Edad: 22
Sexo: F

...Program finished with exit code 0
Press ENTER to exit console.

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

```
class Persona {
public:
```



```
// Atributos
string nombre;
int edad;
char sexo;

Persona(string nombre, int edad, char sexo) : nombre(nombre), edad(edad), sexo(sexo) {}

// Método para solicitar y almacenar los valores de la persona
void solicitarDatos() {
    cout << "Ingrese el nombre: ";
    //getline porque el nombre puede contener espacios.
    getline(cin, nombre);
    cout << "Ingrese la edad: ";
    cin >> edad;
    cout << "Ingrese el sexo (Masculino/Femenino): ";
    cin >> sexo;
    cin.ignore();
}
};

void imprimir(Persona persona_imp) {
    cout << "\nDatos ingresados:" << endl;
    cout << "Nombre: " << persona_imp.nombre << endl;
    cout << "Edad: " << persona_imp.edad << endl;
    cout << "Sexo: " << persona_imp.sexo << endl;
}

int main() {
    Persona persona1("", 0, ' ');
    persona1.solicitarDatos();

    Persona persona2("", 0, ' ');
    persona2.solicitarDatos();

    imprimir(persona1);
    imprimir(persona2);

    return 0;
}
```



2. Redacta una conclusión en la que destagues la importancia y utilidad de la sobrecarga de métodos o funciones.

Mediante esta actividad pude aprender la importancia de la utilidad de la sobrecarga de métodos ya que me permite definir varias versiones en una misma función como si la reciclara pero cambiarle el método para que me de una función distinta a lo que se esta solicitando. Me agrado la organización que me dio ya que la actividad uno nos solicita operaciones similares pero con diferentes datos y no estar creando nuevos nombres para cada punto y únicamente llamar a la función determinada; me pude percatar que puedo yo seguir agregando más funciones sin que modifique el código y eso lo veo como un buen manejo eficiente al momento de agregar o modificar para seguir desarrollando aun mas el programa.

Referencias

Ceballos, J. (2007). Programación orientada a objetos con C++ Haga clic para ver más opciones [Archivo PDF]. Recuperado el 30 de noviembre del 2024, de https://kupdf.net/download/ceballos-programaci-oacute-n-orientada-a-objetos-con-c-4ed_58a8b3476454a7596fb1e8d9_pdf

Clases y Objetos en C++ (Práctica 1) Haga clic para ver más opciones (s.f). Recuperado el 30 de noviembre del 2024, de <https://www.codingame.com/playgrounds/50557/clases-y-objetos-en-c-practica-1/miembros-de-clase-en-c-variables-y-metodos>

ChatGPT. (n.d.). Chatgpt.com. Recuperado el 30 de noviembre del 2024, de <https://chatgpt.com>

TylerMSFT. (n.d.). *Sobrecarga de funciones*. Microsoft.com. Recuperado el 30 de noviembre del 2024, de <https://learn.microsoft.com/es-es/cpp/cpp/function-overloading?view=msvc-170>

