# Comp 4603

# Advanced C++

|  |  |  |  |
| --- | --- | --- | --- |
| Assignment | 9 | Part | 1 |

Student Name: Alisher Shamayev

BCIT ID: A01182685

Task 1: What is MVC?

|  |
| --- |
| MVC stands for Model View Controller, this is an architecting method in developing application, and the idea is divide parts of application into three different parts.  Model is backend or data or logic of application.  View is responsible for UI of application.  Controller is kind of a backend but more of middle pointer between Model and View that does some scripts in example handle user request, or update the page and etc. |

How to implement MVC in C++?

|  |
| --- |
| //  // main.cpp  // Assignment9P1  //  // Created by GUW06 on 2023-03-15.  //  #include <iostream>  **using** **namespace** std;  **class** Model {  **public**:  **void** setModel(**int** x) {  modelData = x;  }  **int** getModel() **const** {  **return** modelData;  }  **private**:  **int** modelData;  };  **class** View {  **public**:  **void** printView(**int** x) {  cout << "Data: " << x << endl;  }  };  **class** Controller {  **public**:  Controller(Model& model, View& view) :  modelData(model), dataView(view) {}  **void** setData(**int** data) {  modelData.setModel(data);  dataView.printView(modelData.getModel());  }  **private**:  Model& modelData;  View& dataView;  };  Basic implementation of MVC in C++, Model class works with the data of application, while View prints the data for us and Controller works with both of Model and a View. |

Task 2: What is a functor in C++?

|  |
| --- |
| It is a class that may act as a function, so which means in example we have Addition class. After lets say it has a function in class which add two integers, so when someone will create an object of class Addition so it actually may be called as a function. |

Task 3: Give a simple code example of a functor

|  |
| --- |
| //  // Functor.cpp  // Assignment9P1  //  // Created by GUW06 on 2023-03-15.  //  #include <iostream>>  #include <stdio.h>  **using** **namespace** std;  **class** Addition {  **public**:  **int** **operator**()(**int** x, **int** y) **const** {  **return** x+y;  }  };  **int** main() {  Addition add;  cout<< add(10,20)<<endl;  **return** 0;  } |

Task 4: What is a lambda in C++?

|  |
| --- |
| Lmabda is unnamed object function which may be functor, or also can be named as anonymous function. The key of lambda expressions is to make a one inline small function in cases when either it is used limited amount of times or smale inline function where object is required. |

Task 5: Give a simple code example of a lambda expression

|  |
| --- |
| //  // lambda.cpp  // Assignment9P1  //  // Created by GUW06 on 2023-03-15.  //  #include <stdio.h>  #include <iostream>  **using** **namespace** std;  **int** main() {  **int** x = 100, y = 788;  **auto** lambdaFunc = [](**int** x, **int** y) { **return** x + y; };  cout<< "Sum of " << x << " and " << y << " is "<<lambdaFunc(x,y)<<endl;  **return** 0;  } |