Contents

[Standard Template Library 2](#_Toc86513620)

[2018 2](#_Toc86513621)

[2017 2](#_Toc86513622)

[2016 2](#_Toc86513623)

[2015 2](#_Toc86513624)

[2014 2](#_Toc86513625)

[Inheritance 4](#_Toc86513626)

[2018 4](#_Toc86513627)

[2017 4](#_Toc86513628)

[2016 5](#_Toc86513629)

[2015 5](#_Toc86513630)

[2014 5](#_Toc86513631)

[Memory 6](#_Toc86513632)

[2018 6](#_Toc86513633)

[2017 7](#_Toc86513634)

[2016 7](#_Toc86513635)

[2015 7](#_Toc86513636)

[2014 7](#_Toc86513637)

[Function Creations 8](#_Toc86513638)

[2018 8](#_Toc86513639)

[2017 8](#_Toc86513640)

[2016 8](#_Toc86513641)

[2015 8](#_Toc86513642)

[2014 8](#_Toc86513643)

# Standard Template Library

### 2018

*Compare the Standard Template Library list container and the Standard Template Library vector container in terms of efficiency when:*

* *Adding/Removing the element in the middle of the container. (4 Marks)*
* *Adding/Removing the element at the end of the container. (4 Marks)*

#### Bank of Europe has a customer base of over 100 million clients. Due to the large number of clients it needs to store client information in a linked list data structure. A client has an account number and a name.

#### Provide a class Client with which instances can be linked forward and backwards to other instances of Client. (6 marks)

#### Implement a method which adds an instance of Client to a linked list of Client. (6 marks)

### 2017

#### Compare the Standard Template Library list container and the Standard Template Library vector container in terms of efficiency when:

#### Adding/Removing the element at the end of the container. (4 Marks)

#### Getting the kth element in the container. (4 Marks)

#### As part of a Role-Playing Game (RPG) you are tasked with creating a data structure which can efficiently handle many inventory items. Inventory items are regularly added/removed to/from a player’s inventory.

#### Provide a class InventoryItem with which instances can be linked forward and backwards to other instances of InventoryItem. (6 marks)

#### Implement a method which removes an instance of InventoryItem from a linked list of InventoryItems. (6 marks)

### 2016

#### Give an example of when it is appropriate to use a list data structure and when it is appropriate to use a vector data structure. (4 marks)

#### Describe the following Standard Template Library containers:

#### Stack (4 marks)

#### Queue (4 marks)

### 2015

### 2014

#### Rent-A-Wreck has many vehicles for rent worldwide. There is high turnover of vehicles every year. Due to the large number of vehicles it needs to store vehicle information in an appropriate data structure. A vehicle has a chassis number and a make.

#### Provide a class Vehicle with which instances can be linked forward and backwards to others instance of Vehicle. (6 marks)

#### Implement a method which adds an instance of Vehicle to the linked list before a given instance of Vehicle. (6 marks)

# Inheritance

### 2018

#### Explain inheritance in relation to C++. (2 Marks)

Inheritance is a mechanism of reusing and extending existing classes without modifying them, thus producing hierarchical relationships between them. Inheritance is almost like embedding an object into a class.

#### Describe the difference between Public, Private and Protected inheritance? (3 Marks)

**Public**

This makes public members of the base class public in the derived class, and the protected members of the base class remain protected in the derived class.

**Protected**

This makes the public and protected members of the base class protected in the derived class.

**Private**

This makes the public and protected members of the base class private in the derived class.

#### Explain, using an example, Polymorphism in C++. (5 Marks)

Polymorphism in C++ means, the same entity (function or object) behaves differently in different scenarios. For example, the `+` operator can perform two specific functions at two different scenarios i.e when the `+` operator is used in numbers, it performs addition. While the same `+` operator is used in the string, it performs concatenation.

int a = 1;

int b = 2;

int sum = a + b;

// sum = 3

string s1 = "Hello ";

string s2 = "World";

string s3 = s1 + s2;

// s3 = Hello World

#### Create a UML diagram for the design of a Game Engine for the following game description. You should also provide a brief description of the engine with your answer.

#### You have been tasked with creating a First-Person Shooter (FPS) where the goal is to eliminate all other Non-Player Characters (NPCs) in an arena. The arena also contains ‘friendly’ characters. If a ‘friendly’ is killed the player is eliminated. The game should have the following features (10 marks)

#### Heads Up Display (HUD)

#### Inventory System

#### Health pick-ups

#### Armour

#### Weapons

#### Player Stats

#### Health

#### Armour

#### Non-Player Characters

#### Enemies

#### Friendly

### 2017

#### In general, if B::f is a function in the base class then a derived can take the actions below. Explain, using an example, each action:

#### Extend B::f (2 Marks)

#### Replace B::f (2 Marks)

#### Inherit B::f (2 Marks)

### 2016

#### Explain the difference between a statically bound function and a dynamically bound function. (2 Marks)

**Static binding**

This happens when all information needed to call a function is available at the compile-time.

**Dynamic binding**

This happens when the compiler cannot determine all information needed for a function call at compile-time.

#### What is a pure virtual function? (2 Marks)

This a function that must be overridden in a derived class and need not be defined.

#### Implement a base class **Body**. Derive classes **RigidBody** and **SoftBody** from **Body**. A **Body** has a mass and a pointer to an array of 2d positions. A **RigidBody** has elasticity and a **SoftBody** has viscosity. Write the class definitions and the constructors for all classes. All data members of your classes should be private. (6 Marks)

### 2015

#### Implement a base class Entity. Derive classes PlayerEntity and AiEntity from Entity. An Entity has an id number, a position in 2d space and a current health. A PlayerEntity has a name and an AiEntity can be visible/invisible. Write the class definitions and the constructors for all classes. All data members of your classes should be private. (6 Marks)

### 2014

#### Write a base class Worker and derived classes HourlyWorker and SalariedWorker. Each worker has a name and salary rate. Define a method computePay(int hours)in the base class Worker that can be overloaded by derived classes. computePay(int hours)computes the weekly pay for every worker. An hourly worker gets paid for the actual number of hours worked, if hours is at most 40. If the hourly worker worked more than 40 hours, the excess is paid at time and a half. The salaried worker gets paid the hourly wage for 40 hours, no matter what the actual number of hours is. (9 Marks)

#### Write a program that fills an array with 6 workers (a mixture of hourly and salaried workers) and computes their pay. (3 Marks)

# Memory

### 2018

#### Memory in C++ is divided into four categories. List and describe each of the four categories. (8 Marks)

**Code**

The instructions for all functions. Makes it possible for re-entrant code. Self-modifying code is usually prohibited.

**Static Data Memory**

Contains Global and static variables. Allocated fixed size when the program is loaded.

**The Run-Time Stack**

Every function allocates an activation record on the run time stack. An activation record will maintain space for the parameter, the return address, space for saved internal registers and other machine information and space for the local variables.

**The Heap Memory**

An area of dynamic memory that is controlled by the program via new and delete

#### Describe the different named variables in the following program and explain in what category of memory each variable resides. If the variable is a pointer detail what memory it points to. (6 Marks)

int clearBuffer(char \* buffer) {

    static int functionCount = 0;

    functionCount++;

    int charactersCleared = 0;

    while (buffer[charactersCleared] != '\0') {

        buffer[charactersCleared++] = ' ';

    }

    return ++charactersCleared;

}

int main() {

    int cleared;

    char \* buffer = new char[50];

    buffer[49] = '\0';

    cleared = clearBuffer(buffer);

    delete buffer;

}

#### What is reference counting in relation to C++? In your answer you should give an example of how reference counting is used. (6 Marks)

### 2017

#### In C++ what is an activation record? (2 Marks)

#### Describe the contents of an activation record. (4 Marks)

#### For the code segment below, show the contents of the runtime stack and value of each variable after:

#### The triangularNumber function has been called for the last time.(6 Marks)

#### The printResult function has been called. (4 Marks)

void printResult(int value) {

    cout << "Final result is " << value << "\n";

}

int triangularNumber(int it) {

    if (it <= 1) {

        return 1;

    } else {

        int num = it \* triangularNumber(it - 1);

        return num;

    }

}

void main() {

    int iterations = 4;

    int result = triangularNumber(iterations);

    printResult(result);

}

#### Explain, using an example, the role of the constructor and destructor of an object. (4 Marks)

### 2016

#### What is a dangling pointer? (2 Marks)

### 2015

### 2014

#### Describe the different named variables in the following program, and explain in what category of memory each variable resides. If the variable is a pointer detail what memory it points to. (6 Marks)

static const int MAX\_SIZE = 256;

bool initialiseArray(int \* arrayParam, int sizeParam) {

    if (size > MAX\_SIZE) {

        return false;

    }

    for (int i = 0; i < sizeParam; i++) {

        arrayParam[i] = 9;

    }

    return true;

}

void main() {

    int \* myArray = new int[30];

    bool res = initialiseArray(myArray, 30);

    delete myArray;

}

# Function Creations

### 2018

*In relation to C++, describe operators and operands. (2 Marks)*

*Describe the function of operator overloading in C++? Provide an example of operator overloading in your answer. (4 Marks)*

*What does a C++ compiler do when it encounters a function template? (2 Marks)*

*Write a function template which compares two parameters of the same type. It should return true if they are equal and false if they are not. (3 Marks)*

*Write a function template that will print an array of any type. (4 Marks)*

*Write a container class that can hold an object of any type. You should provide accessors and modifiers for the contained object. (5 Marks)*

### 2017

### 2016

### 2015

*Describe, using an example, the purpose of a function template. (5 Marks)*

*The function defined below returns the product of the first n elements of an array of integers:*

***int calculateProduct(int data[], int n);***

*Provide an implementation for the function. (2 Marks)*

*Alter the function so that it will work with any object that supports the multiplication operator (\*). (5 Marks)*

*What are the advantages/disadvantages of operator overloading? (2 Marks)*

*Provide the definition and implementation of a class that represents a 3D space vector i.e. a vector has an x, y, and z component. Overload the following operators for your class:*

* *Addition operator(+), to add two vectors. (3 Marks)*
* *Subtraction operator(-), to subtract one vector from another. (3 Marks)*

### 2014

#### Write a function template which will find the largest of three variables of any type. (3 Marks)