

2 part 2

Question 1

Correct

Mark 1.00 out of 1.00

[Flag question](#)

The code fragment given below crashes.

What changes would you do to fix it?

```
#using <System.dll>

using namespace System;
using namespace System::Diagnostics;

int main()
{
    array<Stopwatch^>^ StopwatchList;
    StopwatchList = gcnew array<Stopwatch^>(2);

    StopwatchList[0]->Start();
    Threading::Thread::Sleep(500);
    StopwatchList[0]->Stop();

    return 0;
}
```

StopwatchList = gcnew array<Stopwatch^>(gcnew Stopwatch, gcnew Stopwatch); ✓

The line

```
StopwatchList = gcnew array<Stopwatch^>(2);
```

only allocates space for the handle `StopwatchList` and does not allocate space for each of the `Stopwatch` objects of the array on the managed heap. As such, `StopwatchList[0]` does not point anywhere. Therefore, we need to allocate space for each object using `gcnew Stopwatch` and we must do that for the number of elements we need, which is 2. Then we should put those as array elements enclosed in braces, `({})`.

Question 2

Incorrect

Mark 0.00 out of 1.00

[Flag question](#)

In a multi-threaded implementation, seven threads are running. The bit locations assigned to each thread to identify the bit locations in the `shutdown` and `heartbeat` bytes are listed below.

Thread 1: b0
Thread 2: b1
Thread 3: b2
Thread 4: b3
Thread 5: b4
Thread 6: b5
Thread 7: b6

According to the convention described in the lectures, if the thread management thread (TMT) decides to shutdown the threads 3 and 4, what value should the TMT write to the `shutdown` byte? Enter your answer as a 2-digit hexadecimal number. ✗

Question 3

Correct

Mark 1.00 out of 1.00

[Flag question](#)

As per the approach described in the lectures, if the thread management thread is going to execute 4 threads, the thread barrier object must be initialized to count how many threads at the barrier?

✓

If the thread management thread is starting 4 threads, then those 4 plus the thread management thread itself makes it five threads.

To shut down **threads 3 and 4**, we set the bits b2 and b3 to 1.

Let's build the byte:

- b3 = 1
- b2 = 1
- all others = 0

Binary (bits 7..0): **00001100**.

00001100₂ = **0x0C**.

Correct answer: **0x0C**

Question 4

Correct

Mark 1.00 out of 1.00

Flag question

In a multi-threaded implementation, seven threads are running. The bits assigned to each thread to identify the bit locations in the shutdown and heartbeats bytes are listed below.

Thread#0: b0
Thread#1: b1
Thread#2: b2
Thread#3: b3
Thread#4: b4
Thread#5: b5
Thread#6: b6

According to the convention described in the lectures, as part of heartbeats monitoring if the thread management thread sees the heartbeats byte value as 0x44, what should the TMT thread do?

- A. Reset stopwatches of Thread#2 and Thread#6 and write 0x00 to heartbeats byte. ✓
- B. Reset all except the stopwatches of Thread#2 and Thread#6 and write 0x00 to heartbeats byte.
- C. Reset stopwatches of Thread#4 and Thread#0 and write 0x00 to heartbeats byte.
- D. Reset all except the stopwatches of Thread#4 and Thread#0 and write 0x00 to heartbeats byte.

Your answer is correct.

0x44 = 0b0100 0100. Therefore, bits b2 and b6 are set. This means TMT is satisfied that Thread#2 and Thread#6 are alive and therefore the TMT will reset only the Stopwatches corresponding to these two threads and will write 0x00 to heartbeats byte.

The correct answer is: Reset stopwatches of Thread#2 and Thread#6 and write 0x00 to heartbeats byte.

Question 5

Correct

Mark 1.00 out of 1.00

Flag question

In a multi-threaded implementation, seven threads are running. The bits assigned to each thread to identify the bit locations in the shutdown and heartbeats bytes are listed below.

Thread#0: b0
Thread#1: b1
Thread#2: b2
Thread#3: b3
Thread#4: b4
Thread#5: b5
Thread#6: b6

According to the convention described in the lectures, as part of heartbeats monitoring if Thread#2 sees the heartbeats byte value as 0x43, which bits of the heartbeats byte should Thread#2 set to one?

- A. All bits except threads b4.
- B. All bits except threads b2.
- C. b2 ✓
- D. b4

Your answer is correct.

0x43 = 0b 0100 0011. This indicates that TMT has put the Thread#2 bit (b2) down. Hence, Thread#2 is satisfied that the TMT is alive and it can now put the bit b2 back up to 1 and reset its Stopwatch.

The correct answer is: b2

Question 6

Correct

Mark 1.00 out of
1.00[Flag question](#)

In a multi-threaded implementation, seven threads are running. The bits assigned to each thread to identify the bit locations in the shutdown and heartbeats bytes are listed below.

Thread#0: b0
 Thread#1: b1
 Thread#2: b2
 Thread#3: b3
 Thread#4: b4
 Thread#5: b5
 Thread#6: b6

According to the convention described in the lectures, as part of heartbeats monitoring if Thread#2 sees the heartbeats byte value as 0x44, which bits of the heartbeats byte should Thread#2 set to one?

- A. All bits except threads b2.
- B. all bits
- C. none ✓
- D. b2

Your answer is correct.

0x44 = 0b 0100 0100. This indicates that TMT has not put the Thread#2 bit (b2) down. Hence, Thread#2 is not yet satisfied that the TMT is alive and therefore it will not do anything and will check again next time.

The correct answer is: none

Question 7

Correct

Mark 1.00 out of
1.00[Flag question](#)

The TMT.h file contents are given below

```
#pragma once
using namespace system;
#include "StopwatchObjects.h"

using namespace System::Threading;
using namespace System::Diagnostics;

ref struct ThreadProperties
{
    ThreadStart^ ThreadStart_;
    bool Critical;
    String^ ThreadName;
    uint8_t BitID;

    ThreadProperties(ThreadStart^ start, bool crit, uint8_t bit_id, String^ threadName)
    {
        ThreadStart_ = start;
        Critical = crit;
        Threadname = threadName;
        BitID = bit_id;
    }
};

ref class ThreadManagement
{
public:
    //set up shared memory
    error_state setSharedMemory();
    //start threads
    void threadFunction();
    //process heartbeats
    error_state processHeartbeats();
    //shutdown threads
    void shutdownThreads();
    //get shutdown flag
    bool getShutdownFlag();
private:
    SH_ThreadManagement^ SM_TM_;
    SH_Error^ SH_Error_;
    SH_GRSS^ SH_GRSS_;
    array<Stopwatch^>^ StopwatchList;
    array<Thread^>^ ThreadList;
    array<ThreadProperties^>^ ThreadPropertiesList;
};
```

What is the number of elements you would expect to have in the array pointed to by the StopwatchList handle? In the answers list TMT means Thread Management Thread.

- A. Number of threads the TMT has to manage minus 1.
- B. 1
- C. Number of threads the TMT has to manage + 1.
- D. Equal to the number of threads the TMT has to manage. ✓

Your answer is correct.

The number of Stopwatch objects required is equal to the number of threads the TMT has to look after, as TMT has to keep track of timing related to heartbeats of each thread it is supervising.

The correct answer is: Equal to the number of threads the TMT has to manage.

Question 8

Correct

Mark 1.00 out of
1.00

[Flag question](#)

In a multi-threaded implementation, seven threads are running. The bits assigned to each thread to identify the bit locations in the shutdown and heartbeats bytes are listed below.

Thread#0: b0
Thread#1: b1
Thread#2: b2
Thread#3: b3
Thread#4: b4
Thread#5: b5
Thread#6: b6

According to the convention described in the lectures, if the thread management thread sets the shutdown byte to 0x21, which of the above threads will shutdown?

- A. Thread#0 and Thread#5 ✓
- B. All threads except Thread#1 and Thread#6.
- C. All threads except Thread#0 and Thread#5.
- D. Thread#1 and Thread#6

Your answer is correct.

0x21 = 0b01000001. Therefore, bits b0 and b5 are set. This means the while loops of Thread#0 and Thread#5 will terminate and hence will shutdown.

The correct answer is: Thread#0 and Thread#5

Question 9

Correct

Mark 1.00 out of
1.00

[Flag question](#)

During tele-operation of a mobile vehicle which of the following threads is non-critical?

- A. LiDAR thread
- B. Thread management thread
- C. GNSS Thread ✓
- D. Display thread

Your answer is correct.

Tele-operation means driving the vehicle from afar, in general, without seeing the vehicle. The LiDAR data showing the obstructions in front of the vehicle is displayed for the teleoperator to make sure the vehicle does not collide with an obstacle. Under these circumstances GNSS has no role to play, other than simply informing where the vehicle is. Not knowing this information does not hinder the tele-operation.

The correct answer is: GNSS Thread

Question 10

Correct

Mark 1.00 out of
1.00

[Flag question](#)

The purpose of the thread barrier implementation using Barrier objects is to

- A. Let the thread management thread to cross first
- B. To synchronize all threads and let their second phase to start virtually at the same time. ✓
- C. Let the thread management thread to cross last
- D. Let the highest priority thread to cross the barrier first

Your answer is correct.

The threads that implement a thread barrier in them has two phases. The first phase consists of the program statements of the thread before the statement `SignalAndWait()`. The second phase consists of all the statements after `SignalAndWait()`. In a multi-threaded implementation phase 1 of all threads complete asynchronously and then come to the `SignalAndWait()` function call and wait till all other threads come to their own `SignalAndWait()` statement. When all threads have arrived at their respective `SignalAndWait()` statements, phase 2 of all threads start virtually at the same time.

The correct answer is: To synchronize all threads and let their second phase to start virtually at the same time.

Question 11

Correct

Mark 1.00 out of
1.00

[Flag question](#)

Which of the following is essential for the creation of a thread?

- A. A ThreadStart object. ✓
- B. A Barrier object.
- C. A Stopwatch object.
- D. A Thread list

Your answer is correct.

Must have a `ThreadStart` object which basically specifies the starting address of the thread function.

The correct answer is: A ThreadStart object.

Question 12
Correct
Mark 1.00 out of 1.00
Flag question

```
Referring to the ThreadManagement class definition given in TMT.h below,
```

```
#pragma once
#include "System.h"
#include "SystemObjects.h"

using namespace System;
using namespace System::Threading;
using namespace System::Diagnostics;

ref struct ThreadProperties
{
    ThreadStart* ThreadStart_;
    bool Critical;
    String^ ThreadName;
    uint8_t BitID;

    ThreadProperties(ThreadStart^ start, bool crit, uint8_t bit_id, String^ threadName)
    {
        ThreadStart_ = start;
        Critical = crit;
        ThreadName = threadName;
        BitID = bit_id;
    }
};

ref class ThreadManagement
{
public:
    error_state setupSharedMemory();
    //Thread function
    void heartbeat();
    //Process heartbeats
    error_state processHeartbeats();
    //Thread
    void shutdownThreads();
    //Get the count
    void getSharedConfig();
private:
    SM_ThreadManagement^ SM_TM_;
    SM_Laser^ SM_Laser;
    SM_GNSS^ SM_GNSS;
    array<Stopwatch^>^ StopwatchList;
    array<Thread^>^ ThreadList;
    array<ThreadProperties^>^ ThreadPropertiesList;
};

If we execute the following line in our software (as we have done in our main() function),
ThreadManagement^ myTM = gcnew ThreadManagement();
what would be the value of SM_TM_ immediately after the execution of the above line?
```

- A. A valid memory address on the managed heap.
- B. nullptr ✓
- C. -1
- D. An arbitrary garbage value.

Your answer is correct.

As we have not defined a default constructor, the compiler generated default constructor will be called. It will only allocate space to its member variables and set them all to their default values. As a result SM_PM_ will be set to nullptr.

The correct answer is: nullptr

Question 13
Correct
Mark 1.00 out of 1.00
Flag question

For thread management, as defined in the TMT.h file, what is the use of the data member Critical in the ThreadProperties structure?

- A. If Critical is true, when that thread fails, all threads must be terminated. ✓
- B. If Critical is true for a certain thread, then that thread must be started first.
- C. If Critical is true for a certain thread, then that thread must be terminated first.
- D. If Critical is false for a certain thread, then that thread must be started last.

Your answer is correct.

Critical field is a Boolean value that specifies if the thread is a critical one. Based on whether Critical is true or false, the thread management software will choose to shutdown the whole operation or to restart the failed thread, respectively.

The correct answer is: If Critical is true, when that thread fails, all threads must be terminated.

Question 14
Correct
Mark 1.00 out of 1.00
Flag question

Given the following situations, when would you use a mutex for reading/writing? (all reading and writing below refers to reading and writing to shared memory)

1. Reading/writing heartbeat byte.
2. Reading/writing shutdown byte.
3. Reading/writing laser range data.
4. Reading/writing the GNSS data.

- A. 1, 2, 3 and 4
- B. 3 and 4 only ✓
- C. 3 only
- D. 1 and 2 only

Your answer is correct.

Reading and writing the heartbeat or shutdown bytes happen with a single CPU instruction (hence called atomic) and cannot be interfered by other threads or processes. Therefore, there is no need to use a mutex. On the contrary, the laser data and GNSS data are multibyte. There reading and writing can be interrupted by other threads which can lead to race conditions. Therefore, we need to use mutexes for 3 and 4 above.

The correct answer is: 3 and 4 only

Question 15
Correct
Mark 1.00 out of 1.00
Flag question

Noting that the object class Stopwatch is ref class, the type of the StopwatchList member of the SM_ThreadManagement class can also be of type array<Stopwatch^>^ StopwatchList;

- Select one:
- True
 - False ✓

The Stopwatch class is a ref class. Stopwatch objects can only be instantiated on the managed heap. Objects instantiated on the heap, regardless of native heap or managed heap can only be accessed by native class pointers or managed class handles. Hence the array elements cannot be objects themselves but either pointers (if they are created on the native heap) or must be handles as CLR ref object are only allowed to be instantiated on the managed heap.

The correct answer is: False.

2 part 1

Question 1

Correct

Mark 1.00 out of
1.00[Flag question](#)

If a uint8_t type variable A has the value 89, and a uint8_t type variable Mask has the value 0b00010000, then A&Mask gives the decimal value,

The correct answer is 16

Question 2

Incorrect

Mark 0.00 out of
1.00[Flag question](#)

If a uint8_t type variable A has the value 89, and a uint8_t type variable Mask has the value 0b00010000, then A|Mask gives the decimal value,

Correct answer is 89

Question 3

Incorrect

Mark 0.00 out of
1.00[Flag question](#)

If a uint8_t type variable A has the value 89, and a uint8_t type variable Mask has the value 0b00010000, then A^Mask gives the decimal value,

Correct answer is 73

Question 4

Correct

Mark 1.00 out of
1.00[Flag question](#)

If you execute the following code, what would get printed?

```
#include <iostream>
#include <bitset>
#include <cstdint>

using namespace System;

int main()
{
    uint8_t a = 5;
    std::bitset<8>b{ a };
    std::cout << b << std::endl;
    Console::ReadKey();

    return 0;
}
```

Correct answer is

00000101

Question 5
Correct
Mark 1.00 out of 1.00
[Flag question](#)

If you execute the following code, what would get printed?

```
#include <iostream>
#include <bitset>
#include <cstdint>

using namespace System;

int main()
{
    uint8_t a = 5;
    std::bitset<8>b{ a };
    std::cout << b << 1 << std::endl;
    Console::ReadKey();

    return 0;
}
```

000001011 ✓

The correct answer is
000001011

Question 6
Correct
Mark 1.00 out of 1.00
[Flag question](#)

If you execute the following code, what would get printed?

```
#include <iostream>
#include <bitset>
#include <cstdint>

using namespace System;

int main()
{
    uint8_t a = 5;
    std::bitset<8>b{ a };
    std::cout << (b << 1) << std::endl;
    Console::ReadKey();

    return 0;
}
```

00001010 ✓

The correct answer is
00001010

Question 7
Incorrect
Mark 0.00 out of 1.00
[Flag question](#)

If you execute the following code, what would get printed? (for self assessment you can cut and paste and run code - then make sure you learn!).

```
#include <iostream>
#include <bitset>
#include <cstdint>

using namespace System;

int main()
{
    uint8_t a = 5;
    std::bitset<8>b{ a };

    std::cout << b << " " << b.flip(5) << std::endl;
    Console::ReadKey();

    return 0;
}
```

00000101 00010101 ✗

The correct answer is
00000101 00100101

Question 8

Correct

Mark 1.00 out of
1.00[Flag question](#)

If you execute the following code, what would get printed? (for self assessment you can cut and paste and run code - then make sure you learn!).

```
#include <iostream>
#include <bitset>
#include <cstdint>

using namespace System;

int main()
{
    uint8_t a = 5;
    std::bitset<8>b{ a };

    std::cout << b[0] << " " << b[7] << std::endl;
    Console::ReadKey();

    return 0;
}
```

1 0 ✓

The correct answer is

1 0

Question 9

Correct

Mark 1.00 out of
1.00[Flag question](#)

If you execute the following code, how many binary digits will be printed? (for self assessment you can cut and paste and run code - then make sure you learn!).

```
#include <iostream>
#include <bitset>
#include <cstdint>

using namespace System;

int main()
{
    uint8_t a = 5;
    std::bitset<16>b{ a };

    std::cout << b << std::endl;
    Console::ReadKey();

    return 0;
}
```

16 ✓

The correct answer is

16

Question 10

Incorrect
Mark 0.00 out of 1.00

[Flag question](#)

If the function `void DigitalOutput(uint16_t value)` with value=2398 is used to send data to the real Galil PLC we used in the laboratory, which of the Digital Output high byte (bank) LEDs will light up?

The correct answer is DO8 and DO11

Question 11

Correct
Mark 1.00 out of 1.00

[Flag question](#)

A Galil RIO47142 PLC is used for weather sensing and signalling in a remote airport that is unmanned. The PLC's digital output low byte is used to drive high powered relays that turn 8 lights on and off. When there is windshear, the 8 lights in a row flash as follows to indicate to the approaching aircraft the presence of windshear. Turn the lights corresponding to D00, D02, D04 and D06 ON and lights corresponding to D01, D03, D05, D07 OFF. Then wait for one second and toggle the ON/OFF state of all lights. Then wait for another second. This should run repeatedly in a loop as long as there is windshear.

Assuming we use the `DigitalByteOutput()` function used in the PLC assignment, the code used in the loop is as follows:

```
uint8_t Lights = 0x55;

while(Windshear)
{
    //Lights = ?
    DigitalByteOutput(0,Lights);
    Thread::Sleep(1000);
}
```

Which statement would you use to replace the line

`//Lights = ?`

The correct answer is

`Lights^=0xFF`

```
#include <iostream>
```

```
#include <cstdint>
```

```
int main() {
```

```
    uint16_t value = 2398;
```

```
    std::cout << "Decimal: " << value << "\n";
```

```
    std::cout << "Hex: " << std::hex << value << "\n";
```

```
    return 0;
```

```
}
```

Question 12
Correct
Mark 1.00 out of 1.00
[Flag question](#)

An on/off air conditioning system relies on analogue temperature measurement to run the air conditioner. If you were to use a Galil PLC to automate the operation of the air conditioning system, which of the functions we developed for the PLC assignment would you use?

- A. void AnalogOutput(uint8_t channel, double voltage);
and
 bool DigitalBitInput(uint8_t bit);
- B. bool DigitalBitInput(uint8_t bit);
and
 void DigitalBitOutput(bool val, uint8_t bit);
- C. float AnalogInput(uint8_t channel);
and
 void DigitalBitOutput(bool val, uint8_t bit); ✓
- D. void AnalogOutput(uint8_t channel, double voltage);
and
 void DigitalBitOutput(bool val, uint8_t bit);

Your answer is correct.

The correct answer is:

```
float AnalogInput(uint8_t channel);
and
void DigitalBitOutput(bool val, uint8_t bit);
```

Question 13
Correct
Mark 1.00 out of 1.00
[Flag question](#)

The binary bit pattern of a uint8_t type variable A divided by 2 is,

- A. The binary bit pattern of A shifted right by 1 position ✓
- B. The binary bit pattern of A shifted left by 1 position
- C. A & A where '&' is the bitwise AND operator.
- D. A | A where '|' is the bitwise OR operator.

Your answer is correct.

The correct answer is: The binary bit pattern of A shifted right by 1 position

Question 14
Correct
Mark 1.00 out of 1.00
[Flag question](#)

Two uint8_t type data called HiByte and LoByte contain some values. A Galil PLC requires a variable named Data of uint16_t type value to have the HiByte as the high byte of Data and the LoByte as the low byte of Data. Which of the following statements would you use to form the 16-bit number? The variable Data may contain a garbage value to start with.

- A. Data = (HiByte << 8) | LoByte; ✓
- B. Data = Data | (HiByte << 8) | LoByte;
- C. Data = HiByte | (LoByte >> 8);
- D. Data = (Data & 0x00) | HiByte | (LoByte >> 8);

Your answer is correct.

The correct answer is: Data = (HiByte << 8) | LoByte;

Question 15

Correct

Mark 1.00 out of
1.00[Flag question](#)

The binary bit pattern of a `uint8_t` type variable A multiplied by 2 is,

- A. The binary bit pattern of A shifted right by 1 position
- B. The binary bit pattern of A shifted left by 1 position ✓
- C. A & A where '&' is the bitwise AND operator.
- D. A | A where '|' is the bitwise OR operator.

Your answer is correct.

The correct answer is: The binary bit pattern of A shifted left by 1 position

Question 16

Correct

Mark 1.00 out of
1.00[Flag question](#)

Which of the following statements will toggle the bits of a `uint8_t` type variable A store the new result back in A?

- A. A ^= A;
- B. A=~A; ✓
- C. A | A;
- D. A|= A;

Your answer is correct.

The correct answer is: A=~A;

Question 17

Incorrect

Mark 0.00 out of
1.00[Flag question](#)

if A is `uint8_t` and has the value 5, then A^A gives the same result as

- A. !A
- B. A|A
- C. A&A ✗
- D. ~A

Your answer is incorrect.

The correct answer is: !A

1 part

Brave File Edit View History Bookmarks Profiles Tab Window Help

Self-assessment quiz for Weeks 1 and 2 - PART 1 (marks not counted): Attempt review | UNSW

moodle.telt.unsw.edu.au/mod/quiz/review.php?attempt=20683379&cmid=7949209

UNSW

Grade 8.00 out of 15.00 (53.33%)

Question 1
Incorrect
Mark 0.00 out of 1.00
[Flag question](#)

```
#pragma once

class Point
{
    int X;
    int Y;
public:
    Point() = default;
    Point(int x, int y);
    int GetX();
    int GetY();
    void SetX(int x);
    void SetY(int y);
    Point GetPoint();
    ~Point() {};
};
```

The **Point** class we defined as part of the **RobotArm** project is given above. The data members **int X** and **int Y** can be accessed by,

Select one:

A. Any function defined within the class.

B. Any function defined within the class and any functions within a class derived from the **Point** class. X

C. Any function such as a **main** function that is external to the class.

Your answer is incorrect.

The correct answer is: Any function defined within the class.



Question 3

Correct

Mark 1.00 out
of 1.00 Flag
question

A class definition is given below.

```
#pragma once
#include "framework.h"
#include "Point.h"

class Shapes
{
protected:
    Point P;
    double Orientation;
    int R;
    int G;
    int B;
public:
    Shapes() = default;
    Shapes(Point p);
    Shapes(Point p, double orientation);
    Shapes(Point p, double orientation, int r, int b, int g);
    void Move(int deltaX, int deltaY, double deltaOrientation);
    virtual void Draw(HDC h) = 0;
    virtual ~Shapes() {};
};
```

Which of the following makes this class an abstract class?

Select one:

- A. Shapes() = default;
- B. virtual ~Shapes() {};
- C. virtual void Draw(HDC h) = 0; ✓
- D. protected:

Question 4

Incorrect

Mark 0.00 out
of 1.00 Flag
question

Definitions of two classes **Shapes** and **Circle** are given below.

```
#pragma once
#include "framework.h"
#include "Point.h"

class Shapes
{
protected:
    Point P;
    double Orientation;
    int R;
    int G;
    int B;
public:
    Shapes() = default;
    Shapes(Point p);
    Shapes(Point p, double orientation);
    Shapes(Point p, double orientation, int r, int b, int g);
    void Move(int deltaX, int deltaY, double deltaOrientation);
    virtual void Draw(HDC h) = 0;
    virtual ~Shapes() {};
};

class Circle : public Shapes
{
    double Radius;
public:
    Circle() = default;
    Circle(Point c, double radius, int r, int g, int b);
    void Draw(HDC h);
    void Modify();
    virtual ~Circle() {};
};
```

The function **Modify()** has access to

Select one:

- A. Member functions of the **Circle** class only.
- B. Member functions and data members of the **Circle** class only.
- C. All members of the **Circle** class and all public functions of the **Shapes** class only. X
- D. All members of both classes

Your answer is incorrect.

The correct answer is: All members of both classes

Question 5

Incorrect

Mark 0.00 out
of 1.00 Flag
question

The **Circle** class definition we used in the **RoboticArm** program is given below.

```
#pragma once
#include "Shapes.h"

class Circle : public Shapes
{
    double Radius;
public:
    Circle() = default;
    Circle(Point c, double radius, int r, int g, int b);
    void Draw(HDC h);
    virtual ~Circle() {};
};
```

The keyword **default** in the class definition does the following.

Select one:

- A. It simply tells that there will be a default constructor
- B. It informs the compiler that the body of the default constructor will be provided by us, the programmers.
- C. It requests the compiler to provide the compiler generated default constructor.
- D. It informs the compiler that the default constructor will not have any parameters passed to it.

Your answer is incorrect.

The correct answer is: It requests the compiler to provide the compiler generated default constructor.

Question 6

Incorrect

Mark 0.00 out
of 1.00 Flag
question

Referring to all classes of **RoboticArm** solution, the **Circle** class does not have direct access to the data member P .

P X R G B

Question 7
Correct
Mark 1.00 out of 1.00
[Flag question](#)

The least sensible constructor for the **Line** class we used in the **RoboticArm** solution is,

Select one:

- A. Line(Point p, double length, double orientation, int r, int g, int b);
- B. Line(Point startPoint, double length, double orientation, int r, int g, int b);
- C. Line(Point endPoint, double length, double orientation, int r, int g, int b);
- D. Line(Point startPoint, Point endPoint, double length, double orientation, int r, int g, int b); ✓
- E. Line(Point startPoint, Point endPoint, int r, int g, int b);

Your answer is correct.

The correct answer is:

Line(Point startPoint, Point endPoint, double length, double orientation, int r, int g, int b);

Question 8
Incorrect
Mark 0.00 out of 1.00
[Flag question](#)

Which of the following object instantiations is correct? Drag and drop the correct answer to the space provided.

Shapes *ShapesPtr = new Shapes; x .

Shapes* ShapesPtr = new Circle(Point(10,10), 10,10,10,10);

Shapes *ShapesPtr = new Shapes(Point(10,10));

Shapes MyShape;

Your answer is incorrect.

The correct answer is:

Which of the following object instantiations is correct? Drag and drop the correct answer to the space provided. [Shapes* ShapesPtr = new Circle(Point(10,10), 10,10,10,10).].

Question 9
Correct
Mark 1.00 out of 1.00
[Flag question](#)

Which of the following object instantiations is incorrect? Drag and drop the answer to the space provided Shapes*ShapesPtr = new Shapes; ✓ .

Shapes* ShapesPtr = new Circle; Shapes* ShapesPtr = new Line; Shapes* ShapesPtr = new Link;

Your answer is correct.

The correct answer is:

Which of the following object instantiations is incorrect? Drag and drop the answer to the space provided [Shapes*ShapesPtr = new Shapes;].

Question 10
Correct
Mark 1.00 out of 1.00
[Flag question](#)

As used in the **RoboticArm** solution, which of the **void Draw(HDC h)**; functions is not a polymorphic function?

Select one:

- A. The one declared in the **Shapes** class.
- B. Th one declared in the **Circle** class.
- C. The one declared in the **Line** class.
- D. The one declared in the **Robot** class. ✓

Your answer is correct.

The correct answer is: The one declared in the **Robot** class.

Question 11

Correct

Mark 1.00 out
of 1.00 Flag
question

The **Robot** class declaration is given below.

```
1 #pragma once
2 #include "Point.h"
3 #include "Link.h"
4 #include <iostream>
5 #include <fstream>
6 #include <vector>
7
8 struct LinkProperties
9 {
10     double Length;
11     double Angle;
12     double HubDiameter;
13     int Red;
14     int Green;
15     int Blue;
16 };
17
18 class Robot
19 {
20     Point Base;
21     std::vector<Link*> Links;
22 public:
23     Robot() = default;
24     Robot(Point base, std::vector<LinkProperties> data);
25     void Draw(HDC h);
26     void Move(std::vector<double> angles);
27     ~Robot();
28     friend std::ostream& operator<<(std::ostream& os, const Robot& r);
29 };
```

The declaration given in line #21 means,

Select one:

- A. a one dimensional array containing an unspecified number of **Link** objects.
- B. a one dimensional array containing an already specified number of **Link** objects.
- C. a one dimensional array containing an unspecified number of pointers to **Link** objects. ✓
- D. a one dimensional array containing an already specified number of pointers to **Link** objects.

Your answer is correct.

The correct answer is: a one dimensional array containing an unspecified number of pointers to **Link** objects.

Question 12

Correct

Mark 1.00 out
of 1.00 Flag
question

The **RoboticArm** software will crash if the length of the vector defined in line #26 in the code fragment given below is,

```
1 #pragma once
2 #include "Point.h"
3 #include "Link.h"
4 #include <iostream>
5 #include <fstream>
6 #include <vector>
7
8 struct LinkProperties
9 {
10     double Length;
11     double Angle;
12     double HubDiameter;
13     int Red;
14     int Green;
15     int Blue;
16 };
17
18 class Robot
19 {
20     Point Base;
21     std::vector<Link*> Links;
22 public:
23     Robot() = default;
24     Robot(Point base, std::vector<LinkProperties> data);
25     void Draw(HDC h);
26     void Move(std::vector<double> angles);
27     ~Robot();
28     friend std::ostream& operator<<(std::ostream& os, const Robot& r);
29 };
```

Select one:

- A. Less than the number of joints in the robot arm.
- B. Less than or equal to the number of joints in the robot arm.
- C. Equal to the number of joints in the robot arm.
- D. Greater than the number of joints in the robot arm. ✓

Your answer is correct.

The correct answer is: Greater than the number of joints in the robot arm.

Question 13

Correct

Mark 1.00 out
of 1.00 Flag
question

Considering the code fragment given below, which of the following functions are overloaded?

```
1 #pragma once
2 #include "Point.h"
3 #include "Link.h"
4 #include <iostream>
5 #include <fstream>
6 #include <vector>
7
8 struct LinkProperties
9 {
10     double Length;
11     double Angle;
12     double HubDiameter;
13     int Red;
14     int Green;
15     int Blue;
16 };
17
18 class Robot
19 {
20     Point Base;
21     std::vector<Link*> Links;
22 public:
23     Robot() = default;
24     Robot(Point base, std::vector<LinkProperties> data);
25     void Draw(HDC h);
26     void Move(std::vector<double> angles);
27     ~Robot();
28     friend std::ostream& operator<<(std::ostream& os, const Robot& r);
29 };
```

Select one:

- A. **Robot** function ✓
- B. **Draw** function
- C. **Move** function
- D. **~Robot** function

Your answer is correct.

The correct answer is: **Robot** function

Question 14

Incorrect

Mark 0.00 out
of 1.00 Flag
question

In a certain **Robot** class, we want to use a statement as given below to print all joint angles of the robot, where **UR** is an instance of the **Robot** class.

Robot UR;**std::cout << "The joint angles are " << UR << std::endl;**Select the correct function declaration you would include in the **Robot** class to achieve this.**Select one:**

- A. friend std::ostream& operator<<(std::ostream& os, const Robot& r);
- B. friend std::ostream operator<<(std::ostream& os, const Robot& r);
- C. friend std::ostream& operator<<(std::ostream os, const Robot& r); X
- D. friend std::ostream operator<<(std::ostream os, const Robot& r);

Your answer is incorrect.

The correct answer is: friend std::ostream& operator<<(std::ostream& os, const Robot& r);

Question 15

Incorrect

Mark 0.00 out
of 1.00 Flag
question

Given the following class definition we used for **RoboticArm** project,

```
#pragma once
#include "Line.h"

class Link : public Line
{
    double HubRadius;
public:
    Link() = default;
    Link(Point p, double length, double angle, double radius, int r, int g, int b);
    void Draw(HDC h);
    virtual ~Link() {};
};
```

which of the following functions is NOT inherited by the **Link** class.

Select one:

- A. Circle::Draw(HDC h);
- B. Line::Draw(HDC h);
- C. Shapes::Draw(HDC h); X
- D. Shapes::Move(int x, int y, double deltaOrientation);

Your answer is incorrect.

The correct answer is: Circle::Draw(HDC h);

1 part 2

Question 2

Correct

Mark 1.00 out
of 1.00[Flag
question](#)

A pointer named `IntPtr` to an `int` type object can be declared with

Select one:

- A. `int IntPtr;`
- B. `int* IntPtr;` ✓
- C. `int& IntPtr;`
- D. `int^ IntPtr;`

Your answer is correct.

The correct answer is: `int* IntPtr;`

Question 3

Correct

Mark 1.00 out
of 1.00[Flag
question](#)

In a standard C++ class definition, the constructor allocates memory dynamically. What is the best place to insert code to relinquish the allocated memory.

Select one:

- A. End of the default constructor.
- B. End of all constructors.
- C. End of the main function.
- D. In the destructor. ✓

Your answer is correct.

The correct answer is: In the destructor.

Question 4

Correct

Mark 1.00 out
of 1.00[Flag question](#)

In C++ CLR (common language runtime) a handle named `IntHandle` to an object of type `int` can be declared with,

Select one:

- A. `int IntHandle;`
- B. `int *IntHandle;`
- C. `int &IntHandle;`
- D. `int ^IntHandle;` ✓

Your answer is correct.

The correct answer is: `int ^IntHandle;`

Question 5

Incorrect

Mark 0.00 out
of 1.00[Flag question](#)

In standard C++ we create space for an array of 100 `int` type objects, dynamically using,

Select one:

- A. `int *IntArray = new int[100];`
- B. `int *IntArray[100];`
- C. `int *IntArray = new int(100);` ✗
- D. `int *IntArray = gcnew int(100);`

Your answer is incorrect.

The correct answer is: `int *IntArray = new int[100];`

Question 6

Incorrect

Mark 0.00 out
of 1.00[Flag
question](#)

In managed code a handle (also called a reference variable - similar to a pointer in unmanaged C++) named `IntArray` to 100 `int` type objects can be dynamically created by,

Select one:

- A. `array<int> ^IntArray = gcnew array<int>(100);`
- B. `array<int^> ^IntArray = gcnew array<int>(100);` X
- C. `array<int> IntArray = gcnew array<int>(100);`
- D. `array<int> ^IntArray = new array<int>(100);`

Your answer is incorrect.

The correct answer is: `array<int> ^IntArray = gcnew array<int>(100);`

Question 7

Incorrect

Mark 0.00 out
of 1.00[Flag
question](#)

In standard C++ if `IntArray` is pointing to an area in heap memory which contains 100 `int` type objects, the memory is released by,

Select one:

- A. `if (IntArray != NULL)
delete [] IntArray;`
- B. `if (IntArray != NULL)
delete IntArray;`
- C. Memory is automatically released, nothing needs to be done. X

Your answer is incorrect.

The correct answer is: `if (IntArray != NULL)
delete [] IntArray;`

Question 8

Incorrect

Mark 0.00 out
of 1.00Flag
question

In C++ (CLI) the dynamically allocated memory referenced by `IntArray` is released by.

Select one:

- A. Memory is automatically released, nothing needs to be done.
- B. `if(IntArray != NULL)` ×
`delete IntArray;`
- C. `if(IntArray != NULL)`
`delete [] IntArray;`

Your answer is incorrect.

The correct answer is: Memory is automatically released, nothing needs to be done.

Question 9

Correct

Mark 1.00 out
of 1.00Flag
question

A function declaration is given below.

`double XToThePowerY(double x, double y);`

A pointer named `FcnPtr` that can point to this function can be declared by;

Select one:

- A. `double (*FcnPtr)(double, double);` ✓
- B. `double *(FcnPtr)(double, double);`
- C. `double *FcnPtr(double, double);`
- D. `double FcnPtr(double, double);`

Your answer is correct.

The correct answer is: `double (*FcnPtr)(double, double);`

Question 10

Incorrect

Mark 0.00 out
of 1.00[Flag question](#)

A code fragment is given below.

```
int main()
{
    std::cout << main << std::endl;
    return 0;
}
```

What will happen if we execute this code?

Select one:

- A. The `main` function will be executed recursively and the system will crash. X
- B. Will not be able to run because it will not compile.
- C. The `main` function will be executed recursively but the system will not crash.
- D. Will print on the screen the `main` function's address in memory.

Your answer is incorrect.

The correct answer is: Will print on the screen the `main` function's address in memory.

Question 11

Correct

Mark 1.00 out
of 1.00 Flag
question

A **ref** type class will always be instantiated on,

Select one:

- A. Managed heap. ✓
- B. Either managed heap or the native heap.
- C. Native heap.
- D. Stack.

Your answer is correct.

The correct answer is: Managed heap.

Question 12

Incorrect

Mark 0.00 out
of 1.00 Flag
question

A value type object in C++ CLR can be instantiated on,

Select one:

- A. Stack only
- B. Stack and managed heap
- C. Managed heap only
- D. Stack and native heap ✗

Your answer is incorrect.

The correct answer is: Stack and managed heap

Question 13

Correct

Mark 1.00 out
of 1.00 Flag
question

It is undesirable to pass large objects to functions because a copy of the passed object will be put on the stack and, the time the stack operation takes and the stack space it consumes makes the code very inefficient.

Select one:

- True ✓
- False

The correct answer is 'True'.

Question 14

Correct

Mark 1.00 out
of 1.00 Flag
question

One way to avoid copying large objects on to the stack when passing large objects as parameters to functions is,

Select one:

- A. Pass the object by reference. ✓
- B. Pass the object by value.
- C. Create the object on the heap and pass the reference to the object to the function, instead the object itself.

Your answer is correct.

The correct answers are: Pass the object by reference., Create the object on the heap and pass the reference to the object to the function, instead the object itself.

Question 15

Correct

Mark 1.00 out
of 1.00[Flag
question](#)

One of the differences between **value struct** and **value class** is,

Select one:

- A. value class has its members private by default and the value struct has its members public by default. ✓
- B. value class has its members public by default and the value struct has its members private by default.
- C. value class has its members public by default and the value struct has its members public by default.
- D. value class has its members private by default and the value struct has its members private by default.

Your answer is correct.

The correct answer is: value class has its members private by default and the value struct has its members public by default.

Question 16

Incorrect

Mark 0.00 out
of 1.00 Flag
question

An event handler implementation is given below.

```
#using <System.dll>

using namespace System;
using namespace System::Timers;

void OnElapsed(System::Object^ sender, System::Timers::ElapsedEventArgs^ e);
int main()
{
    Timer^ MyTimer = gcnew Timer(500);
    MyTimer->AutoReset = true;
    MyTimer->Elapsed += gcnew System::Timers::ElapsedEventHandler(&OnElapsed);
    while(!Console::KeyAvailable);
    MyTimer->Stop();
    Console::CursorVisible = true;

    return 0;
}
void OnElapsed(System::Object^ sender, System::Timers::ElapsedEventArgs^ e)
{
    Console::CursorVisible = false;
    Console::SetCursorPosition(20, 2);
    Console::WriteLine(DateTime::Now);
}
```

This code failed to run the even handler. The reason is,

Select one:

- A. A **Console::.ReadKey();** statement is missing.
- B. A **while(!Console::KeyAvailable);** statement is blocking the execution.
- C. **MyTimer->Enabled = true;** is missing.
- D. **MyTimer->Start();** is missing.

Your answer is incorrect.

The correct answer is: **MyTimer->Start();** is missing.

Question 17

Correct

Mark 1.00 out
of 1.00 Flag
question

A code fragment is given below.

```
1 #using <System.dll>
2
3 using namespace System;
4 using namespace System::IO;
5
6 value struct GNSS
7 {
8     double Latitude;
9     double Longitude;
10    GNSS(double lat, double lng) :Latitude(lat), Longitude(lng) {}
11 };
12
13 int main()
14 {
15     String^ Filename = "GNSS.m";
16     StreamWriter^ sw = gcnew StreamWriter(Filename);
17     array<GNSS>^ GNSSData = { GNSS(1,2),GNSS(3,4),GNSS(5,6),GNSS(7,8)};
18     for (int i = 0; i < GNSSData->Length; i++)
19     {
20         sw->WriteLine("{0, 10:F3} {1, 10:F3}", GNSSData[i].Latitude, GNSSData[i].Longitude);
21     }
22     sw->Close();
23     //Wait for keypress
24     Console::ReadKey();
25     return 0;
26 }
```

If line #6 is changed to **ref struct GNSS** what changes would be needed in line #17?**Select one:**

- A. array<GNSS^>^ GNSSData = { gcnew GNSS(1,2),gcnew GNSS(3,4),gcnew GNSS(5,6),gcnew GNSS(7,8) };
- B. array<GNSS>^ GNSSData = { gcnew GNSS(1,2),gcnew GNSS(3,4),gcnew GNSS(5,6),gcnew GNSS(7,8) };
- C. array<GNSS^> GNSSData = { GNSS(1,2),GNSS(3,4),GNSS(5,6),GNSS(7,8) };
- D. array<GNSS^>^ GNSSData = {GNSS(1,2),GNSS(3,4),GNSS(5,6),GNSS(7,8) };

Your answer is correct.

The correct answer is: array<GNSS^>^ GNSSData = { gcnew GNSS(1,2),gcnew GNSS(3,4),gcnew GNSS(5,6),gcnew GNSS(7,8) };