

This paper contains 18 questions. MCQ questions 1 to 15 carry 2 marks each, and there is no negative marking. Answer on the attached answer sheet and insert the answer sheet in your answer book. Fill in your details on the answer sheet.

Calculators are not allowed.

Answer in ink. Do not use red ink.

This paper contains 5 pages including the MCQ sheet

**Q1:** Given the declarations `int a = 35; int &b = a; int *c = &a;` which of the following will insert the value 70 into `a`? Choose the best answer.

- a) `b = 2*a;`                      b) `*c = 2*a;`              c) `a = 2*b;`  
d) `a *= 2;`                      e) all of these

**Q2:** Given the code `int var = 3; int *ptr = &var; int &ref = *ptr;` which of the following precisely doubles the value stored in `ref` and in `var`? Choose the best answer.

- a) `var = var * 2;`              b) `ref = 2 * var;`                      c) `var = *ptr * 2;`  
d) `ref = &var * 2;`              e) `a, b and c are correct`

**Q3:** When data and functionality are combined into a class, we call this process of combining...

- a) inheritance                      b) polymorphism                      c) abstraction  
d) encapsulation                      e) data overloading

**Q4:** The scope of a private member-variable is...

- a) automatic scope      b) object scope      c) class scope      d) function scope      e) block scope

**Q5:** A lambda that is intended to modify variables that belong to the containing function, will use...

- a) `(&)`              b) `(=)`              c) `[&]`                      d) `[=]`                      e) `(*)`

**Q6:** Which one of these is correct ...

- a) arguments are variables      b) arguments are values                      c) Parameters are values  
d) Parameters are arguments      e) none are correct

**Q7:** Which of the following is subject to late binding?

- a) Virtual functions                      b) operator overloading                      c) function overloading.  
d) implicit specialisation                      e) explicit specialisation

**Q8:** Which of the following can have identical function signatures

- a) overloaded functions                      b) overloaded methods                      c) virtual functions  
d) overloaded operators                      e) none of these

**Q9:** Class B inherits a member function from class A; and class B is not allowed to share that inherited member with anybody. It is not clear if the member was public or protected in A. What type of inheritance does B use when it inherits from A?

- a) protected
- b) public
- c) multiple
- d) private
- e) none of these

**Q10:** Which type of binding takes place late in the program life-cycle?

- a) dynamic binding
- b) early binding
- c) compile-time binding
- d) static binding
- e) none of these

**Study the following and answer the questions below:**

The following code fragment instantiates and uses a class that inherits from two base classes. The main program simply demonstrates limited use of the object. Several sections of code are missing. These are indicated by markers.

```
#include <iostream>
/**missing code – Q14**/
using namespace std;

class person {
/**missing code – Q11**/:
    string name;
    long SAID;
};

class student /**missing code – Q12**/{
/**missing code – Q11**/:
    long studentID;
    string degree;
};

class myStudent /**missing code – Q13**/{
private:
    float finalMark;
public:
    float getFinalMark(long secretKey) {
        if (secretKey == 12345678) { return finalMark; }
    }
    void setFinalMark(long secretKey, float newMark) {
        if (secretKey == 12345678) { finalMark=newMark; }
    }
};

int main(){
    myStudent Tom;
    //the following is a code fragment from main...
    Tom.SAID = 87654321;
    Tom.setFinalMark(12345678, 75.4);
    cout << "Final mark is now " << Tom.getFinalMark(12345678) << endl;
    //...
    return 0;
}
```

**Q11:** What is missing at marker **/\*\*missing code – Q11\*\*/**

- a) private
- b) public
- c) protected
- d) static
- e) nothing missing here

**Q12:** What is missing at marker **/\*\*missing code – Q12\*\*/**

- a) : public person
- b) : private person
- c) : protected person
- d) : public myStudent
- e) nothing missing here

**Q13:** What is missing at the two markers called **/\*\*missing code – Q13\*\*/**

- a) : public person, public student
- b) : private person, private student
- c) : private student
- d) : protected person, protected student
- e) : protected student

**Q14:** What is missing at marker **/\*\*missing code – Q14\*\*/**

- a) #include <string.h>
- b) #include <string>
- c) #include <inheritance>
- d) #include <polymorphism>
- e) Nothing is missing here

**Q15:** What type of inheritance do we have in this code?

- a) Single public
- b) Multilayer
- c) Hierarchical
- d) Multilevel
- e) Multiple

---

**Question 16:**

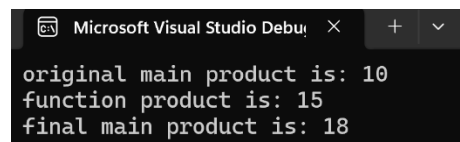
**[6]**

Consider the given application and console output. Convert the function to a named lambda function and convert the function call if needed. Do not change any of the program logic. Your application should produce the same output as the original.

```
#include <iostream>
using namespace std;
void fn(int& val1,int& val2) {
    cout << "function product is: " << (val1++) * (++val2) << endl;
};

int main() {
    int val1 = 5;
    int val2 = 2;

    cout << "original main product is: " << val1 * val2 << endl;
    fn(val1,val2);
    cout << "final main product is: " << val1 * val2 << endl;
    return 0;
}
```



```
Microsoft Visual Studio Debug Console
original main product is: 10
function product is: 15
final main product is: 18
```

**Question 17:****[14]**

Write a class template that does the following:

- It stores a numeric input of any numeric data type, in a member variable called `num`.
- It uses a member function called `numSquared()` to return the square of that value. The returned data type is the same as input data type.
- The member function must be declared in the class and defined outside the class.

Also provide a simple main program that demonstrates the class with integer and float data types. All the code is in one source file.

---

**Question 18:****[50]**

A regular polygon is a closed multisided object where all sides have the same length. Some simple examples are an equilateral triangle, a square and an octagon with equal sides. Using the following abstract class, derive classes for the equilateral triangle, square, and the regular octagon. Call these `regularTriangle`, `regularSquare`, and `regularOctagon`.

```
class regularPolygon {
public:
    float sideLength;
    float NumSides;
    virtual float perimeter() = 0;
    virtual float area() = 0;
};
```

Note: Assuming that the side is  $x$ ,

- The area of an equilateral triangle may be expressed as:  $\frac{\sqrt{3}}{4}x^2$ ,
- And the area of the regular octagon may be expressed as:  $2(1 + \sqrt{2})x^2$ .
- The default length of a side is 1.
- Each class should also insert the appropriate default number of sides.
- Default member values should be configured in the default constructor

Write the entire application in one file. The main program must ask the user to enter the length of the regular polygon's sides. Then using the three derived classes, it must then display in a user-friendly manner, the perimeter and areas of each of the three polygons. You may not modify the abstract class in any way.

**MCQ Answer sheet** – Fill in your details below***Detach this page and return it in your answer book******Clearly delete one of the tables – we will only mark one table***

Student number:

Seat Number:

Signature:

Answer table

| Question | Solution |
|----------|----------|
| 1        |          |
| 2        |          |
| 3        |          |
| 4        |          |
| 5        |          |
| 6        |          |
| 7        |          |
| 8        |          |
| 9        |          |
| 10       |          |
| 11       |          |
| 12       |          |
| 13       |          |
| 14       |          |
| 15       |          |

Spare table

| Question | Solution |
|----------|----------|
| 1        |          |
| 2        |          |
| 3        |          |
| 4        |          |
| 5        |          |
| 6        |          |
| 7        |          |
| 8        |          |
| 9        |          |
| 10       |          |
| 11       |          |
| 12       |          |
| 13       |          |
| 14       |          |
| 15       |          |