

Name: Nashir Jannochan ID: 1325605

True/False: Write out either True or False.

1. True If you do not define one the compiler will create an implicit constructor for your class.
2. True You can define more than one constructor for a class.
3. False When defining an instance of a class, member variables in that class are always initialized to known values.
4. False You can define more than one destructor for a class.
5. True Pointers in C++ are data types that store addresses.
6. False In C++ arrays are just pointers.
7. False Suppose you have an int pointer called ptr pointing to an integer array. C++ understands $*(ptr) + 2$ to mean "move two integers down and retrieve that value".

Multiple Choice:

8. d Abstract Data Types consist of:
 - a.) Data Structures.
 - b.) Algorithms.
 - c.) An Interface.
 - d.) All mentioned.
9. d Which of these best describes the concept of 'Encapsulation'?
 - a.) Hides data and algorithms.
 - b.) Using single interface for general class of actions.
 - c.) Reduce Complexity.
 - d.) All mentioned.
10. a If a class's data members are private, what can we do to access them from the class object?
 - a.) Create public member functions to access those data members.
 - b.) Create private member functions to access those data members.
 - c.) Create protected member functions to access those data members.
 - d.) Private data members can never be accessed from outside the class.
11. b What is the difference between struct and class in C++?
 - a.) All members of a structure are public and structures don't have constructors and destructors.
 - b.) Members of a class are private by default and members of struct are public by default. When deriving a struct from a class/struct, default access-specifier for a base class/struct is public and when deriving a class, default access specifier is private.
 - c.) All members of a structure are public and structures don't have virtual functions.
 - d.) All mentioned.
12. c What is the output of the following program?

Name: Nashir Jannohamed

ID: 1325665

```
#include<iostream>
using namespace std;
class Point {
public:
    Point() { cout << "Constructor called" << endl; }
};

int main() {
    Point t1, *t2;
    return 0;
}
```

- a.) Compiler Error.
- b.) Constructor called
Constructor called
- c.) Constructor called
- d.) Nothing will print.

13. b You must include a class's header file when.

- a.) Use the class as a parameter to a function.
- b.) Use any of that class's member functions.
- c.) Declare a pointer or reference to that class.
- d.) All mentioned.

Short Answer:

14. For this problem, you will be asked to write some code to accomplish a particular task given the code fragment below. Each task will depend on the tasks that came before it. Your code must be syntactically correct.

```
int i1 = 100;
int i2 = -8;
int i3 = 15;

int *p1, **p2, ***p3;
```

- a.) Set p1 to point to i1.

$p1 = \&i1;$

- b.) Using p2 change the value of i1 to the value of i2.

$p2 = \&p1; \quad **p2 = i2$

- c.) Using p3 make p1 point to i3

$p3 = \&p2; \quad ***p3 = \&i3;$

- d.) What does the following code output? `cout << *&*&*&p2;`

15

Name: Nashir Janmohamed

ID: 1325605

15. This following program is supposed print 30 20 10, but it does not. Find all the bugs and show fixed corrected version of the program. Note that `cout<<"30 20 10";` is not a correct fix.

```
int main(){
    int arr[3] = { 5, 10, 15 };
    int* ptr = arr;

    *ptr = 10;           // set arr[0] to 10
    *(ptr + 1) = 20;     // set arr[1] to 20
    ptr += 2;
    ptr[0] = 30;         // set arr[2] to 30
    ptr++;
    while (ptr > arr) {
        ptr--;
        cout << ' ' << *ptr;    // print values
    }
    cout << endl;
} #include <iostream>
using namespace std;
```

```
int main() {
    int arr[3] = { 5, 10, 15 };
    int *ptr = arr;
    *ptr = 10;
    *(ptr + 1) = 20;
    ptr += 2;
    ptr[0] = 30;
    ptr++;
    while (ptr > arr) {
        ptr--;
        cout << ' ' << *ptr;
    }
    cout << endl;
}
```


Name: *Nashir Janmohamed*

ID: *1325605*

16. Suppose you're tasked with implementing a coin counter program for a digital piggy bank. The incomplete class declaration is as follows:

```
class CountingPiggy
{
public:
    CountingPiggy ();
    void addDollar();    // Add a dollar coin to the machine.
    void addQuarter();   // Add a quarter to the machine.
    void addDime();      // Add a dime to the machine.
    void addNickel();    // Add a nickel to the machine.
    void addPenny();     // Add a penny to the machine.
    void giveMeBills();  // Print what the user will get.
                        // See the example below.

private:
    // TODO: Add private variables here.
    int m_cents;

};
```

The class's usage may look like:

```
CountingPiggy cp;
cp.addDollar();           // Currently $1.00.
cp.addQuarter();          // Currently $1.25.
for (int i = 0; i < 10; i++) // Add 10 dimes, total becomes $2.25.
    cp.addDime();
for (int i = 0; i < 32; i++) // Add 32 pennies, total becomes $2.57.
    cp.addPenny();
cp.giveMeBills();         // Produces the following output.
```

Output:
2 dollar bill(s)
2 quarter(s)
0 dime(s)
1 nickel(s)
2 penny(ies)

Implement all the member functions so that the machine works as intended. Do not add or modify any of the public members. You are free to add any private members as needed. It may be useful to recall that in C++ the modulus operator is % which gives you the remainder in a division operation.

Name: Nashir Jannmohamed

ID: 1325605

```
#include "CountingPiggy.h"
```

```
#include <iostream>
```

```
using namespace std;
```

```
CountingPiggy::CountingPiggy() { m_cents = 0; }
```

```
void CountingPiggy::addDollar() { m_cents += 100; }
```

```
void CountingPiggy::addQuarter() { m_cents += 25; }
```

```
void CountingPiggy::addDime() { m_cents += 10; }
```

```
void CountingPiggy::addNickel() { m_cents += 5; }
```

```
void CountingPiggy::addPenny() { m_cents += 1; }
```

```
void CountingPiggy::giveMeBills() {
```

```
    int cents = m_cents;
```

```
    int dollars = cents / 100;
```

```
    cents %= 100;
```

```
    int quarters = cents / 25;
```

```
    cents %= 25;
```

```
    int dimes = cents / 10;
```

```
    cents %= 10;
```

```
    int nickels = cents / 5;
```

```
    cents %= 5;
```

```
    //print
```

```
    cout << dollars << " dollar bill(s)\n";
```

```
    cout << quarters << " quarter(s)\n";
```

```
    cout << dimes << " dime(s)\n";
```

```
    cout << nickels << " nickel(s)\n";
```

```
    cout << cents << " penny(ies)" << endl;
```

```
}
```