Quiz 07

Due May 23 at 11:59pm **Points** 7 **Questions** 7
Available until May 23 at 11:59pm **Time Limit** None

Instructions

Answer the following questions.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	16 minutes	7 out of 7

Score for this quiz: **7** out of 7
Submitted May 23 at 5:25pm
This attempt took 16 minutes.

Question 1

1 / 1 pts

Match the definitions.

Correct!

Multiple functions with the same name, but different input/output types and independent definitions

function overloading

Correct!

Multiple functions with the same name, but different input/output types and

function templates

similar definitions

Question 2

1 / 1 pts

In a function template one or more type parameters are used that can be replaced with arbitrary types.

Correct!

● True

○ False

Question 3

1 / 1 pts

Assume we want to have function **id()** that returns the input of any type, e.g,

- id(5) returns **5**,

- id("hi") returns **"hi"**,

- id('@') returns **'@'**, etc.

What would be the correct function template for this purpose?

○

```
typename<template T>
T id (T x) {
   return x;
}
```

○
```
typename<T>
T id (T x) {
   return x;
}
```

Correct!

●
```
template<typename T>
T id (T x) {
   return x;
}
```

Question 4

1 / 1 pts

Class templates are used ...

Correct!

● to avoid defining multiple redundant classes that only differ in data types.

○ when the class has multiple overloading functions.

○ when class functions are overriding a base class's functions.

Question 5

1 / 1 pts

Class **Eq** includes to functions **isEqual()** and **isInequal()** that checks if the two inputs are equal/not-equal, e.g.,

- isEqual(5, 5) returns **true**,

-isInequal('c', 'c') returns **false**,

-isInequal("hi", "hello") returns **true**, etc.

What would be the correct class template for this purpose?

Correct!

●
```
template <typename T>
class Eq {
   public:
      isEqual(T, T);
      isInequal(T, T);

      ....
};
```

○
```
typename <T>
class Eq {
   public:
      isEqual(T, T);
      isInequal(T, T);

      ....
};
```

○
```
typename <template T>
class Eq {
   public:
      isEqual(T, T);
      isInequal(T, T);

      ....
};
```

Question 6

1 / 1 pts

In the previous question, what would be the correct syntax to define function **isEqual()**.

○
```
bool Eq<T>::isEqual(T a,T b) {
   return a == b ;
}
```

Correct!

●
```
template <typename T>
bool Eq<T>::isEqual(T a,T b) {
   return a == b ;
}
```

○
```
template <typename T>
bool isEqual(T a,T b) {
   return a == b ;
}
```

Question 7

1 / 1 pts

In class **Eq** discussed previously, how can you call **isEqual(5, 5)**?

○
```
Eq<int> isEqual(5,5);
```

Correct!

●
```
Eq<int> eqInt;
eqInt.isEqual(5,5);
```

○
```
isEqual<int> (5,5);
```

Quiz Score: **7** out of 7